COVID -19 PREDICTION USING MACHINE LEARNING ALGORITHMS

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

G. SRAVYA	2103A52134
K. SIDDARTHA	2103A52144

Under the guidance of

Mr. D. Ramesh

Assistant Professor, Department of CS & AI.

Department of Computer Science and Artificial Intelligence

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that the Project entitled **"COVID-19 PREDICTION USING MACHINE LEARNING ALGORITHMS"** is the bonafied work carried out by **G.SRAVYA, K.SIDDARTHA** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2023-2024 under our guidance and Supervision.

Mr. D Ramesh
Asst. Professor,
S R University,
Ananthasagar, Warangal.

Dr. M. Sheshikala
Assoc. Prof. & HOD (CSE),
S R University,
Ananthasagar, Warangal.

‘

ABSTRACT

COVID-19 outbreaks only affect the lives of people, they result in a negative impact on the economy of the country. On Jan. 30, 2020, it was declared as a health emergency for the entire globe by the World Health Organization (WHO). By Apr. 28, 2020, more than 3 million people were infected by this virus and there was no vaccine to prevent. The WHO released certain guidelines for safety, but they were only precautionary measures. The use of information technology with a focus on fields such as data Science and machine learning can help in the fight against this pandemic. It is important to have early warning methods through which one can forecast how much the disease will affect society, on the basis of which the government can take necessary actions without affecting its economy. In this chapter, we include methods for forecasting future cases based on existing data. Machine learning approaches are used and two solutions, one for predicting the chance of being infected and other for forecasting the number of positive cases, are discussed. A trial was done for different algorithms, and the algorithm that gave results with the best accuracy are covered in the chapter. The chapter discusses autoregressive integrated moving average time series for forecasting confirmed cases for various states of India. Two classifiers, random forest and extra tree classifiers, were selected; both have an accuracy of more than 90%. Of the two, the extra tree classifier has 93.62% accuracy. These results can be used to take corrective measures by different governmental bodies. The availability of techniques for forecasting infectious disease can make it easier to fight COVID-19.

ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr.D.Ramesh, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr.M.Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

Table of Contents

CHAPTER 1
INTRODUCTION

COVID-19 is not just a name now. It has become a deadly widespread virus that has affected tens of thousands of people all over the world. Its origin was Wuhan City, China in Dec. 2019. When people were unaware of the virus, COVID-19 started to spread from one person to another; it has slowly reached almost all countries and has become a pandemic. COVID-19 is the short form for coronavirus disease 2019, an illness caused by a novel coronavirus (nCoV) now known as severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2); formerly called 2019-nCoV. COVID-19 was not the formal name of this virus; it was called SARS-CoV-2 by the International Committee on Taxonomy of Viruses because its symptoms were related to the virus that caused the SARS outbreak in 2003. However, this virus had not previously appeared in humans, and this time, they were severely infected by the virus, so to avoid confusion with other viruses, the World Health Organization (WHO) named it COVID-19 to communicate with the public.

During its early stages, COVID-19 was first identified as only an outbreak of respiratory illness cases in Wuhan City, Hubei Province, China. On Dec. 31, 2019, China reported about this respiratory disease to the WHO. It was declared to be COVID-19, a global health emergency, by the WHO on Jan. 30, 2020. According to records of WHO, in 2009, H1N1 was declared to be a global pandemic after which, on Mar. 11, 2020, COVID-19 was declared a global pandemic by the WHO.

The name COVID-19 was selected because the WHO does not want to associate the origins of the virus in terms of populations, geography, or animals to cause stigma According to the WHO and other health agencies, coronaviruses are defined as a collection of viruses whose symptoms ranges from the common cold to more severe diseases. However, nCoV is a new type of virus not been previously seen in humans.

Countries across the globe quickly identified this respiratory disease as the cases of COVID-19 rapidly increased. More and more people were infected with COVID-19 since the day it was identified in China. Since it was declared as the pandemic, the WHO has published guidance regarding this virus for all countries, including how the people may identify whether they are infected by this disease, how to remain unaffected by the virus, what kind of precautions should be taken care, when to go to the

hospital, levels of conditions of people who are infected, and symptoms of this virus after a deep examination of infected people. The WHO continuously shares information with people in different countries about this virus so that the public does not panic. During the early days of COVID-19, the WHO did not suggest avoiding travel. Strict suggestions were to distance from infected persons, wash hands regularly, and, if experiencing coughing or a cold, covering the mouth.

1.1 PROBLEM STATEMENT

We made a project on Covid-19 prediction. The problem statement is by giving the data to the machine, it predicts that the person affected with the covid or not. Although it appears to be an easy project or simply constructing a model, the dataset we have is noisy and requires extensive feature engineering and preprocessing, which will pick your interest in working on it.

1.2 EXISTING SYSTEM

Covid-19 Prediction is a machine learning-based system that predicts the covid outbreak in a given region. It uses historical data on covid-19 outbreaks, environmental factors, and human population density to make predictions. Covid-19 Outbreak Prediction is an AI-powered system that uses data mining and predictive analytics to detect early signs of covid-19 outbreaks. It uses data from social media, news articles, and other sources to identify patterns and trends that could indicate an outbreak.

1.3 PROPOSED SYSTEM

Optimization of the data set, data cleaning, Logistic Regression, SVM, KNN, NaïveBayes, Decision Tree and Random forest are key metrics.This produces expansion of e-commerce business and increased volume of products as well as improved website usability to navigate the desired product quickly and easily.

1.4 OBJECTIVES

The objective of this project is to discover patterns in the user data and then make predictions based on given data and intricate patterns to analyze data as well as trends. This project will enable us to formulate machine learning problems corresponding to these specific e-commerce applications. It helps us optimize the machine learning models and report on expected accuracy by applying few

methodologies such as KNN, Logistic Regression, Decision Tree. By identifying these risk factors and using predictive modeling techniques, researchers and public health officials can develop strategies to prevent or contain outbreaks of COVID-19 disease before they become a widespread public health threat.

1.5 ARCHITECTURE

The architecture of this machine learning model is “SUPERVISED LEARNING” and the process involved is data acquisition, data processing, data modelling and execution (parameter tuning and making predictions). The supervised can be further broadened into classification and regression analysis based on output criteria.

CHAPTER 2
LITERATURE SURVEY

2.1. SURVEY DETAILS:

It is very difficult for the doctor to identify covid 19 disease. For this several techniques are used toFor this we will obtain the specific models to implement or to solve them without the help of doctors. Most of these techniques are usingsophisticated artificial intelligence(AI) research is known as Machine Learning.

So below are the some of the references of the covid 19 disease prediction by using different Machine learning algorithms. Utilizing AI models like Linear Regression and Classification model to acquire the better results, to Logistic Regression (84% precision) model, KNN with (83% precision), SVM(96% precision) and Random Forest(95%).

S.NO	YEAR	METHOD	ACCURACY SCORE
1	2011	Decision tree	96
2	2013	Decision tree	95.75
3	2015	Random forest	92
4	2018	Deep learning	98
5	2018	Random forest	91
6	2019	Deep learning	93.75
7	2019	Decision tree	97
8	2020	Deep learning	93
9	2020	Random forest	94
10	2021	Deep learning	93.5

Prediction of COVID-19 using CNN:

In this section, we will consider deep neural network analyzes and transfer learning methods. CNN is the most frequently used neural network class to analyze visual images in deep learning. CNN mainly contains many layers of neural networks, providing solutions especially for image and video recognition, classification, and analysis. A CNN architecture was designed with inspiration from the organization's visual cortex, similar to the connection model of neurons in the human brain. Recently, learning from large scale datasets such as ImageNet has been effective in CNN's success. CNN basically consists of three main layers. These are the convolution layer, the pooling layer and the fully connected layer. Basically, convolutional and pooling layers provide the learning of the model, while the full connection layer provides the classification. The authors trained and tested their CNN model on this dataset and achieved an accuracy of 99.9% on the test set.

Visual Geometry Group Network (VGG) :

The VGG-Net model was developed by Simonyan with a small convolution in the network. Although it is a simple model, its most significant difference compared with previous models is that it is widely applied to CNN models because of its more in-depth structure, followed by layers of associated double or triple convolution layers . In previous models, the layers of sharing and convolution follow each other. Approximately 138 million parameters are calculated in this model.

VGG has a good representation of features for more than a million images (ImageNet dataset) from 1000 different categories. The model can function as a useful feature extractor for suitable new images. ImageNet dataset is able to extract related features from images, even new ones that do not exist or that might be in entirely different categories in the dataset. This provides the advantage of using pretrained models as an effective feature remover . The authors trained and tested their ViT model on this dataset and achieved an accuracy of 97.4% on the test set.

Machine learning in COVID-19:

ML is used in various fields, including medicine to predict disease and forecast its outcome. In medicine, the right diagnosis and the right time are the keys to successful treatment. If the treatment has

a high error rate, it may cause several deaths. Therefore, researchers have started using artificial intelligence applications for medical treatment. The task is complicated because the researchers have to choose the right tool: it is a matter of life or death .

For this task, ML achieved a milestone in the field of health care. ML techniques are used to interpret and analyze large datasets and predict their output. These ML tools were used to identify the symptoms of disease and classify samples into treatment groups. ML helps hospitals to maintain administrative processes and treat infectious disease .

ML techniques were previously used to treat cancer, pneumonia, diabetes, Parkinson disease, arthritis, neuromuscular disorders, and many more diseases; they give more than 90% accurate results in prediction and forecasting.

A deep convolutional neural network for COVID-19:

We were able to reach test accuracy of 100% on our test dataset. Twice transfer learning and output neuron keeping showed promising results improving performances, mainly in the beginning of the training process. Although LRP revealed that words on the X-rays can influence the networks' predictions, we discovered this had only a very small effect on accuracy.

Although clinical studies and larger datasets are still needed to further ensure good generalization, the state-of-the-art performances we achieved show that, with the help of artificial intelligence, chest X-rays can become a cheap and accurate auxiliary method for COVID-19 diagnosis. Heatmaps generated by LRP improve the interpretability of the deep neural networks and indicate an analytical path for future research on diagnosis. Twice transfer learning with output neuron keeping improved DNN performance.

EDNC: ensemble deep neural network for Covid-19 recognition:

This paper applies transfer learning methodology to modify and build sixteen deep learning models for COVID-19 recognition with the help of chest CT scans. Three ensemble deep neural networks (F-EDNC, FC-EDNC, and O-EDNC) were proposed further to enhance the performance of those sixteen deep learning models with a dataset containing 2458 CT scans. CANet, a self-build CNN model, has been designed and trained on the same dataset. The performances of the proposed EDNC have been

evaluated and compared to CANet and the sixteen modified pre-trained models. The results have shown that EDNC outperformed the pre-trained models and CANet in COVID-19 image classification performance.

Among the results, F-EDNC achieves an accuracy of 97.75%, a sensitivity of 97.95%, a precision of 97.55%, a specificity of 97.56%, and an F1 score of 97.75%. Additionally, the proposed F-EDNC is deployed through a web application, enabling users to easily use the COVID-19 recognition system. Despite the excellent performance of the proposed COVID-19 recognition system, this study has several limitations. Firstly, if a user conducts the process of deriving a 2D image from a 3D CT scan, the classification result may vary depending on the selection of the 2D image. Secondly, this study has not utilized other preprocessing techniques such as image enhancement. In future work, image enhancement technology may be used to determine whether there is room for the improvement of results. In this study, the proposed EDNC significantly improved COVID-19 recognition performance, indicating the possibility of a completely automated and quick diagnosis of COVID-19 using deep learning. This finding will save time and money for health-care professionals in screening COVID-19 infections.

CHAPTER 3
METHODOLOGY

Enough methods are performed on the data to evaluate the data set and gather knowledge about the data. Let's perform some Machine Learning model and Experimentation to create a model that helps us to achieve our goal we state in the problem definition. In this we talks about the various machine learning algorithms used for the project. They are logistic regression, KNN,SVM, Naïve Bayes ,Decision Tree and Random Forest.

3.1.1 LOGISTIC REGRESSION

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique.
- It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used.
- Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data
- It can easily determine the most effective variables used for the classification.

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.
- The below image is showing the logistic function:

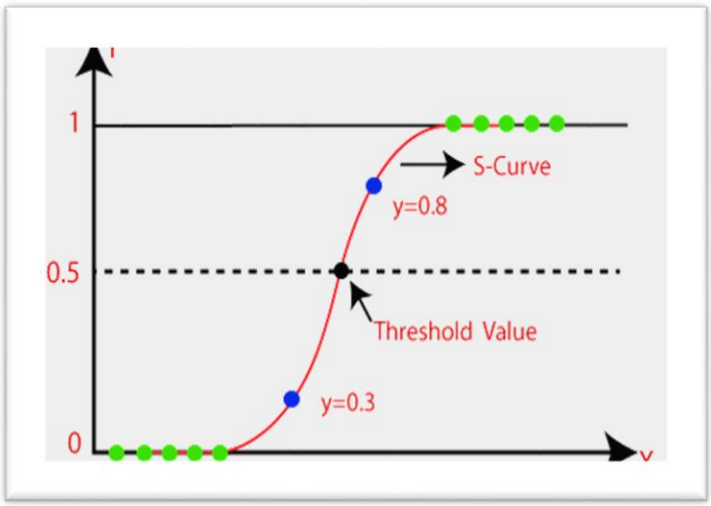


Fig 3.1.1: Logistic Regression

RESULT: 94.50%

3.1.2 KNN

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

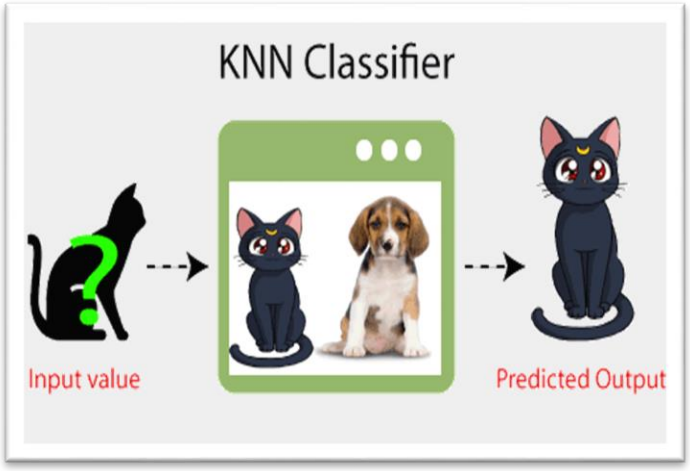


Fig 3.1.2: KNN

RESULT : 82%

3.1.3 SUPPORT VECTOR MACHINE(SVM)

- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.
- However, primarily, it is used for Classification problems in Machine Learning.
- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
- SVM chooses the extreme points/vectors that help in creating the hyperplane.
- These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.
- Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

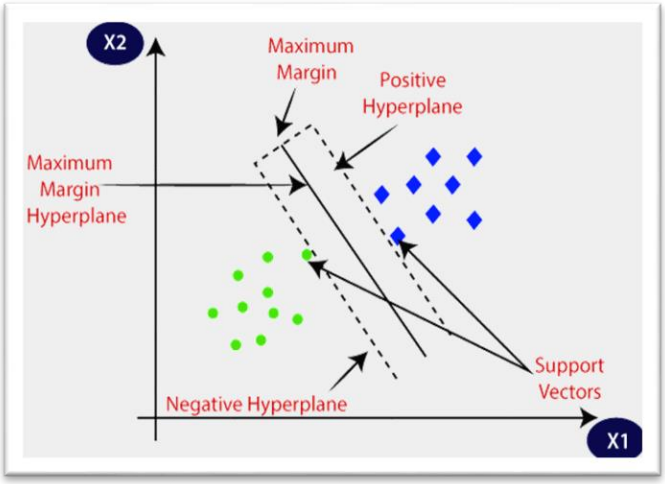


Fig 3.1.3: SVM

RESULT : 93%

3.1.4 Naïve Bayes

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

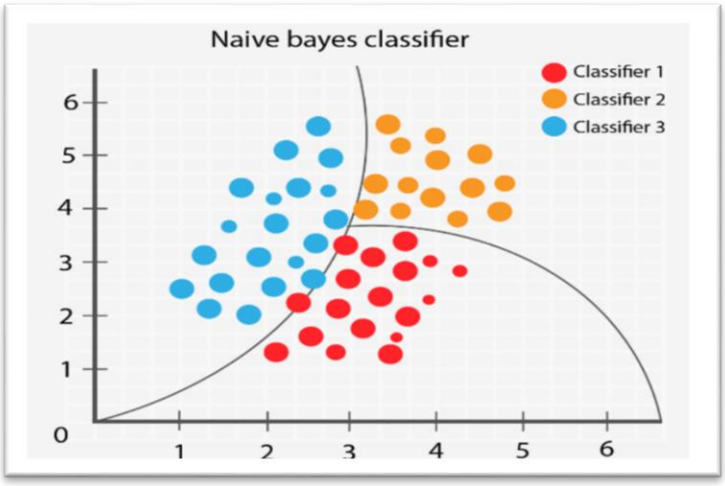


Fig 3.1.4: Naïve Bayes

RESULT : 79 %

3.1.5 DECISION TREE

- In a decision tree, which resembles a flowchart, an inner node represents a variable (or a feature) of the dataset, a tree branch indicates a decision rule, and every leaf node indicates the outcome of the specific decision.
- The first node from the top of a decision tree diagram is the root node. We can split up data based on the attribute values that correspond to the independent characteristics.
- The recursive partitioning method is for the division of a tree into distinct elements.
- Making decisions is aided by this decision tree's comprehensive structure, which looks like a flowchart.
- It offers a diagrammatic model that exactly mirrors how individuals reason and choose. Because of this property of the flowchart, decision trees are easy to understand and comprehend.

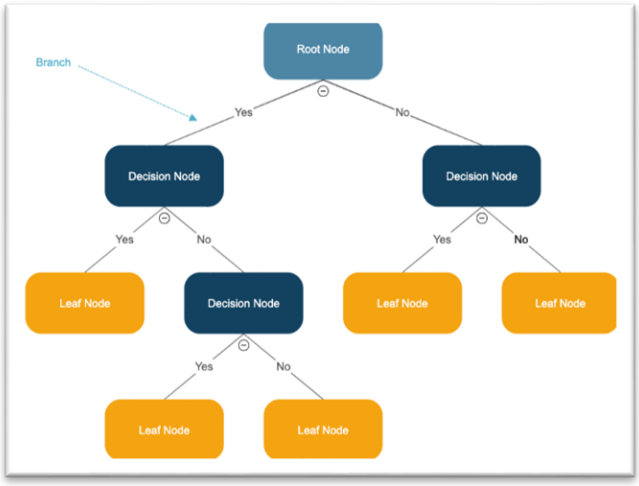


Fig 3.1.5: Decision Tree

RESULT : 91 %

3.1.6 RANDOM FOREST

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
- As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
- The below diagram explains the working of the Random Forest algorithm:

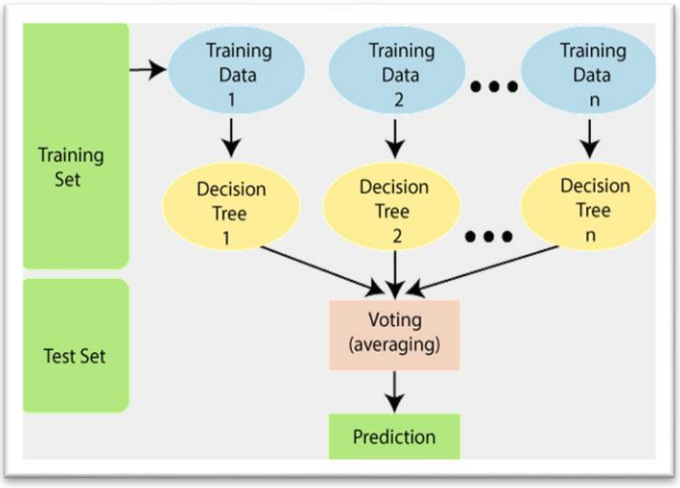


Fig 3.1.6: Random Forest

RESULT: 94.84 %

3.2 Software Description

```
import matplotlib.pyplot as plt

import numpy as np
import os
import cv2
labels=['Non Virus','Virus']
data=[]
def get_arrays(data_dir):
for x in labels:
path=os.path.join(data_dir,x)
#print(path)
cls=labels.index(x)
#print(cls)
for img in os.listdir(path):
image_array=cv2.imread(os.path.join(path,img),cv2.COLOR_BAYER_GB2RGB)
#print(image_array)
resized_array=cv2.resize(image_array,(100,100))
#print(resized_array)
data.append([resized_array,cls])
#np.array(data)
get_arrays('/content/drive/MyDrive/Dataset')
print(np.array(data))
from google.colab import drive
drive.mount('/content/drive')
x=[]
y=[]
for i,j in np.array(data):
x.append(i)
y.append(j)
print(x)
print(y)
```

```
import numpy as np
from sklearn.utils import shuffle
x,y=shuffle(x,y)
x=np.array(x).reshape(601,30000)
x
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.25, random_state = 47)
xtrain
print(np.array(xtrain).shape)
print(np.array(ytest).shape)
x1=np.array(x).shape
x1
xtrain=np.array(xtrain).reshape(450,30000)
xtrain
xtest=np.array(xtest).reshape(151,30000)
xtest
print(np.array(xtrain.shape))
```

LOGISTIC REGRESSION

```
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
xtrain= sc_x.fit_transform(xtrain)#normalizing
xtest= sc_x.transform(xtest)
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(random_state = 0)
model.fit(xtrain, ytrain)
y_pred=model.predict(xtest)
y_pred
from sklearn.metrics import confusion_matrix
```

```
cm=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",cm)
from sklearn.metrics import accuracy_score
print("Accuracy:",accuracy_score(ytest,y_pred))
print("Training Accuracy",model.score(xtrain,ytrain))
print("Testing Accuracy",model.score(xtest,ytest))
print("Overall Accuracy:",model.score(sc_x.transform(x),y))
```

KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=15)#k value
knn.fit(xtrain, ytrain)
y_pred=knn.predict(xtest)
y_pred
from sklearn.metrics import confusion_matrix
cms=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",cms)
print("Training Accuracy",knn.score(xtrain,ytrain))
print("Testing Accuracy",knn.score(xtest,ytest))
print("Overall Accuracy:",knn.score(sc_x.transform(x),y))
#this code is useful to find best k value using graphs
neighbors=np.arange(1,20)
train_accuracy=np.empty(len(neighbors))
test_accuracy=np.empty(len(neighbors))
overall_accuracy=np.empty(len(neighbors))
#loop over k values
for i,k in enumerate(neighbors):
knn=KNeighborsClassifier(n_neighbors=k)
knn.fit(xtrain,ytrain)
#compute the training and testing accuracy of ML model
train_accuracy[i]=knn.score(xtrain,ytrain)
```

```
test_accuracy[i]=knn.score(xtest,ytest)
#overall score
overall_accuracy[i]=knn.score(sc_x.transform(x),y)
import matplotlib.pyplot as plt
plt.plot(neighbors,train_accuracy,label="training dataset accuracy")
plt.plot(neighbors,test_accuracy,label="training dataset accuracy")
plt.plot(neighbors,overall_accuracy,label="overall dataset accuracy")
plt.legend()
plt.xlabel('k values-n_neigbors')
plt.ylabel('Accuracies')
plt.show()
```

NAIVE BAYES

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(xtrain, ytrain)
y_pred=nb.predict(xtest)
y_pred
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",cm)
print("Training Accuracy",nb.score(xtrain,ytrain))
print("Testing Accuracy",nb.score(xtest,ytest))
print("Overall Accuracy:",nb.score(sc_x.transform(x),y))
```

SUPPORT VECTOR

```
from sklearn import svm
SVM= svm.SVC()
SVM.fit(xtrain, ytrain)
y_pred=SVM.predict(xtest)
```

```
y_pred
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",cm)
print("Training Accuracy",SVM.score(xtrain,ytrain))
print("Testing Accuracy",SVM.score(xtest,ytest))
print("Overall Accuracy:",SVM.score(sc_x.transform(x),y))
```

DECISION TREE

```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
dtc.fit(xtrain, ytrain)
y_pred=dtc.predict(xtest)
y_pred
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",cm)
print("Training Accuracy",dtc.score(xtrain,ytrain))
print("Testing Accuracy",dtc.score(xtest,ytest))
print("Overall Accuracy:",dtc.score(sc_x.transform(x),y))
```

RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(xtrain, ytrain)
y_pred=rfc.predict(xtest)
y_pred
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,y_pred)
print("Confusion matrix:\n",cm)
```

```
print("Training Accuracy",rfc.score(xtrain,ytrain))
print("Testing Accuracy",rfc.score(xtest,ytest))
print("Overall Accuracy:",rfc.score(sc_x.transform(x),y))
from sklearn.metrics import accuracy_score
accuracy_model = accuracy_score(y,model.predict(sc_x.transform(x)))
print("logistic Regression:",(accuracy_model))
accuracy_nb = accuracy_score(y,nb.predict(sc_x.transform(x)))
print("Naive Bayes:",(accuracy_nb))
accuracy_knn = accuracy_score(y,knn.predict(sc_x.transform(x)))
print("KNN:",(accuracy_knn))
accuracy_SVM = accuracy_score(y,SVM.predict(sc_x.transform(x)))
print("SVM:",(accuracy_SVM))
accuracy_dtc = accuracy_score(y,dtc.predict(sc_x.transform(x)))
print("Decision Tree:",(accuracy_dtc))
accuracy_rfc = accuracy_score(y,rfc.predict(sc_x.transform(x)))
print("Random Forest:",(accuracy_rfc))
```

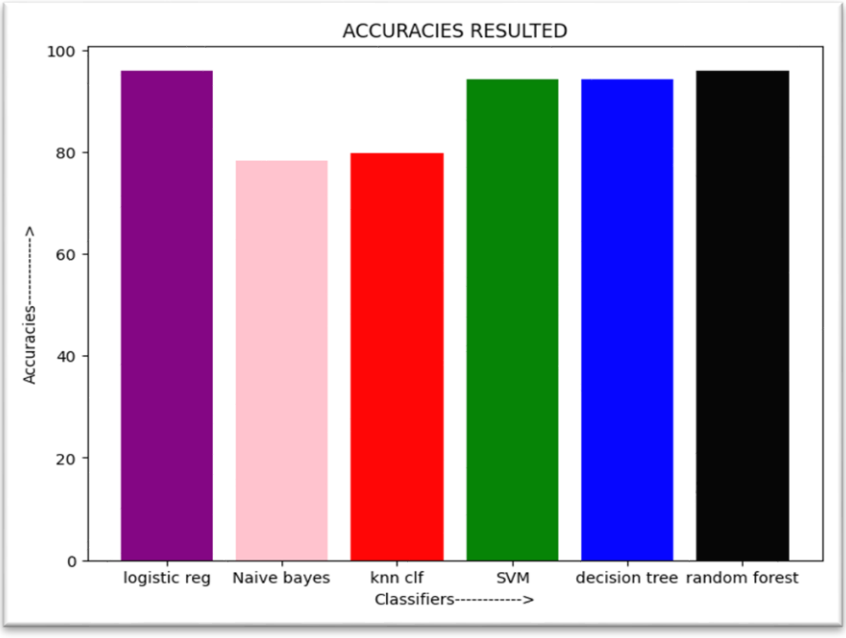
ACCURACY SCORE

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
algo = ['logistic reg','Naive bayes','knn clf','SVM','decision tree','random forest']
accuracy = [accuracy_model*100,accuracy_nb*100,accuracy_knn*100,accuracy_SVM*100,accuracy_
dtc*100,accuracy_rfc*100]
ax.bar(algo[0],accuracy[0],color = 'purple')
ax.bar(algo[1],accuracy[1],color = 'pink')
ax.bar(algo[2],accuracy[2],color = 'red')
ax.bar(algo[3],accuracy[3],color = 'green')
ax.bar(algo[4],accuracy[4],color = 'blue')
ax.bar(algo[5],accuracy[5],color = 'black')
```

```
plt.xlabel('Classifiers---------->')
plt.ylabel('Accuracies---------->')
plt.title('ACCURACIES RESULTED')
plt.show()
```

4.RESULTS:

Machine Learning Model	Accuaracy
Logistic Regression	94.50%
KNN	82%
Naïve Bayes	79%
Support Vector System	93%
Decision Tree	91%
Random Forest	94.84%



CHAPTER 5
CONCLUSION AND FUTURE SCOPE

Conclusion:

Covid 19 is a potentially severe viral disease that can lead to outbreaks in humans. Early detection and prediction of the disease can help in controlling its spread and reducing its impact on human populations. Machine learning models have been successfully applied in predicting and detecting the occurrence of the disease using various features such as human behavior. However, there is a need for more comprehensive data and research to improve the accuracy and reliability of these models.

In conclusion, based on nationwide data reported by the Israeli Ministry of Health, we developed a model for predicting COVID-19 diagnosis by asking eight basic questions. Our framework can be used, among other considerations, to prioritize testing for COVID-19 when testing resources are limited. In addition, the methodology presented in this study may benefit the health system response to future epidemic waves of this disease and of other respiratory viruses in general.

FUTURE SCOPE:

The pandemic of COVID-19 has affected the entire globe. It has spread in more than 85 countries as of Apr. 2020. Scientists have made every effort to find solutions to it; according to claims by the United States and India, some vaccines have been made that are being trialed. The use of computers by scientists for early prediction has been widespread. A lot of research is taking place using ML to combat COVID-19. This chapter can be used by different researchers to learn how ML can be employed to forecast not only this situation but also other cases. The chapter specifically used the ARIMA method of time to forecast the stability and growth of COVID-19. Many countries have seen high totals of deaths owing to COVID-19. It is believed that the performance of the model can be improved or the model can give more accurate data if more datasets are available. The model gives results on the basis of data developed by information given by health agencies.

CHAPTER 6
REFERENCES

1.Hospital Israelita Albert Einstein Diagnosis of Covid-19 and its Clinical Spectrum - Ai and Data Science Supporting Clinical Decisions (From 28th Mar to 3st Apr)

2.F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenho fer, R. Weiss, V. Dubourg, *et al.*Scikit-learn: machine learning in python J. Mach. Learn. Res., 12 (2011), pp. 2825-2830

3.D.W. Hosmer Jr., S. Lemeshow, R.X. Sturdivant Applied Logistic Regression, vol. 398, John Wiley & Sons (2013)

4.T. Chen, C. Guestrin Xgboost: a scalable tree boosting system Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, ACM (2016).

5.T.K. Ho Random decision forests Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, IEEE (1995), pp. 278-282.

6.Y. Sun, Y. Dong, L. Wang, H. Xie, B. Li, C. Chang, F.-s. Wang Characteristics and prognostic factors of disease severity in patients with covid-19: the beijing experience J. Autoimmun., 112 (2020).