
AI In Medical Domain

| SHAIKH ABDULLAH

abdullah042973@gmail.com

Table of Contents

ABSTRACT	3
1. INTRODUCTION.....	3
I. BACKGROUND AND MOTIVATION.....	3
II. OVERVIEW	6
III. RESEARCH GOALS AND APPROACH.....	7
2. LITERATURE REVIEW	8
3. ARCHITECTURE AND DESIGN.....	9
I. DESIGN STRATEGY	9
II. DATA ANALYSIS.....	10
III. PARAMETRIC ANALYSIS OR SENSITIVITY, AND UNCERTAINTY ANALYSIS.....	11
4. METHODOLOGY AND PROPOSED SOLUTION	13
I. MODELLING.....	13
a. Algorithms.....	13
b. Loss Function	21
c. Optimizer	22
II. PERFORMANCE EVALUATIONS	23
5. VALIDATION OF MODELLING AND RESULT	26
6. CONCLUSION	28
7. REFERENCES.....	29

Coronavirus Disease Prediction Using Machine Learning

SHAIKH ABDULLAH KHAWAJA

E-mail: abdullah042973@gmail.com

Department of Information Technology, MVLU College, Mumbai University, INDIA

ABSTRACT

This paper refers to machine learning in the medical domain. At the end of 2020, we all are facing a global pandemic due to coronavirus. There is a total of 146 million active cases of coronavirus and 3.1 million peoples already lost their lives and still counting. I observed and study coronavirus is spreading from one another and it takes a symptom to appear in 5 to 6 days. Then I realize we don't have perfect medicine yet and it's hard to recognize COVID-19 patience during the incubation period. so, I ended up with the idea is to create a machine learning model that predicts COVID-19 patience risk. Studying various classification algorithms, I selected the best fit models(algorithm) using supervised machine learning technology that trains on given datasets and predicts the outcome probability or result. Then compare the accuracy, loss, and optimization of all trained models.

Keywords: Classification; Artificial Intelligence; Machine Learning; Deep Learning; Bias and Variance; Confusion Matrix; Medical Domain;

1. Introduction

I. Background and motivation

In North America, late 1920s first coronavirus infection occurred in chickens. Sir Arthur Schalk and Merle Hawn in 1931 made the first detailed report on respiratory infection of chicken in North Dakota which is in the U.S. Because of this disease 40-90% of chicks were infected [1]. Leland David Bushnell and Carl Alfred Brandly recognized the virus that caused the infection in 1933 [2]. The virus was known as infectious bronchitis virus (IBV). Charles D. Hudson and Fred Robert Beaudette cultured the virus first time in 1937 [3]. Two more animals were infected by coronaviruses that caused brain disease in them and mice developed hepatitis disease also called as mouse hepatitis virus (MHV). These three diseases are related to each other and the virus is similar in all but acts differently in different animals. There is some coronavirus common in animal-like feline CoVs that is usually found in cats and causes the neurological disorder, Canine CoVs found in pigs and cats and causes diarrhoea, Porcine CoVs found in rodent but it was originally from the bat, etc. There is total 28 types of coronavirus.

First human coronavirus in human was detected in the mid-1960s [4]. There are four main groups of coronaviruses [5, p. 371–383] [5] [5, p. 371–383], known as alpha, beta, gamma, and delta. Common

human coronavirus is 1.229E (alpha coronavirus), 2. NL63 (alpha coronavirus), 3. OC43 (beta coronavirus), 4. HKU1 (beta coronavirus). Other life-threatening coronaviruses in human are 5. MERS-CoV (the beta coronavirus that causes Middle East respiratory syndrome, or MERS), 6. SARS-CoV (the beta coronavirus that causes severe acute respiratory syndrome, or SARS), 7. SARS-CoV (the novel coronavirus that causes coronavirus disease 2019 or COVID-19). Sometimes coronavirus evolves among animals and can infect also leading to their death. Its mutation and environment adaptiveness ability allow it to evolve in different environment. Three recent examples of this evolved virus are COVID-19, SARS-CoV, and MERS-CoV.

What is AI (Artificial Intelligence)?

Let's divide word into parts. Artificial means manmade it could be anything that is built by a human like a fan, computer, machine, etc. and Intelligence means the capability of solving problems by own. A basic idea behind Artificial Intelligence is that a machine that can solve its problem by own or a machine can solve a given problem in its own correct way just like a human does. In simpler term you can also say that a machine that mimics human for example nowadays we have Siri, Alexa, Google Assistant and many more they all are AI who communicate with human just like us. AI term first introduced by John McCarthy in 1956 [6].

What is machine learning? Machine learning is a subset of AI basically it is a technology that solves problems in a Mathematical ways ML is used in Data mining and many other algorithms fall under Machine learning as per our needs. The term 'machine learning' was coined in 1959 by Arthur Samuel [7], An American IBMer and pioneer in the field of Computer Gaming and Artificial Intelligence. How does it work? The process of learning begins with data. Data helps a machine to understand the problem or situation. The idea of machine learning is computers learn automatically without human interruption or explicitly programmed.

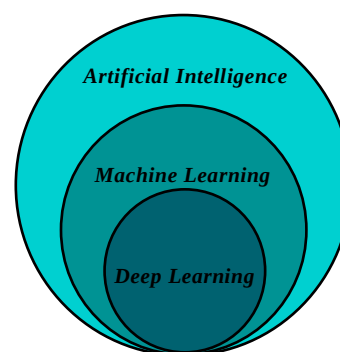


Fig 1.1 AI, ML, and DL relation

There are three types of machine learning **1** supervised machine learning: supervised learning is a type of machine learning where we teach the machine using labelled data.

2 unsupervised machine learning: in this, we feed unstructured data and the machine will figure out the pattern of data this type of learning is usually used for creating Clusters by checking their similar properties like colour, shape, size, etc. **3** reinforcement learning: In this, Machine will learn from its past experiences. If machine prediction is right, it will get the reward and if not then got a penalty this is a tactic used for training a pet.

Deep Learning

As the above diagram shows that Deep Learning is a subset of Machine Learning. So why Deep Learning? Because of the feature selection. in machine learning, we decide what feature should machine used for training, and Machine Learning is not capable of solving complex problems, image classification and it required well-structured data. Deep learning was introduced to the machine learning community by Rina Dechter [8].

Deep Learning inspired by neurons (neurons is a brain cell that specialized in transmitting information to one another nerve cell, muscle, and gland) In the brain there are billions of neurons connected each

other in order to transmit information [9]. When neurons receive or send information, they transmit an electrical signal which is carried by another neuron until it reaches its destination to Cerebrum. The cerebrum is a large outer part of the brain that is responsible for thinking, learning, reading, emotions, speech, muscle movement, and also for other senses like vision and hearing.

ANN (Artificial Neural Network) is a computer algorithm that solves a problem using neural networks just like our brain neurons does (transmission of data). It is fully connected to each other and solve the problem using mathematical procedure. As shown below

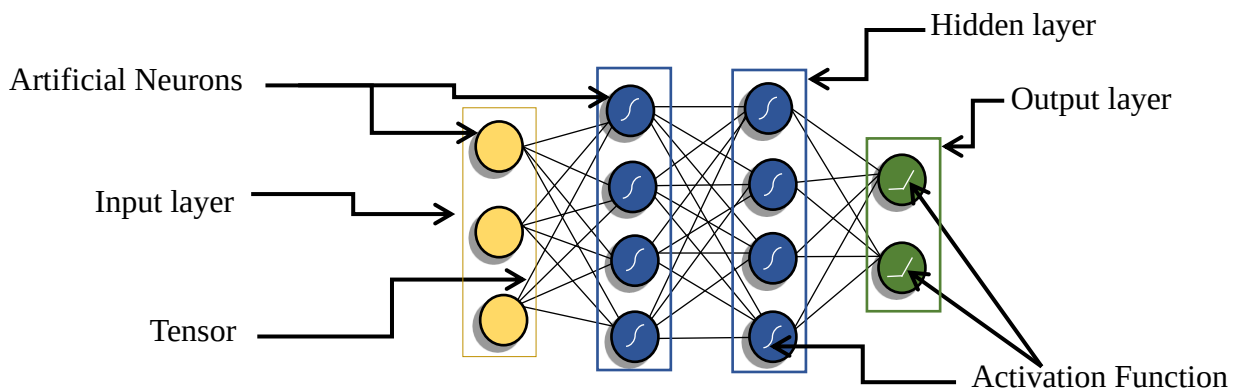


Fig 1.2 Basic understanding of Neural Network

The above diagram is a basic structure of ANN but in reality, this is a way more complex structure. The diagram consists of a 3-part input layer, hidden layer, and output layer. Input layer takes input like datasets, images, random numbers, etc then each neuron of the input layer divides the input into small features and transfers the information through Tensor to hidden layers neurons. Tensor contains weight (weight is nothing but the recipe of generating much like output). Hidden layer neurons activate according to the activation function (activation function allows us to fit non-linear data) main use of activation function is when to activate a particular neuron-like in the above example, I am using the sigmoid function which is denoted by $\sigma(x) = 1/(1+e^{-x})$ (“-x” replace by weight which is sent by input layer neurons). Sigmoid activation function or s-curved function activate on 1 or greater than .5 and finally activated neurons pass to the output layer.

Let’s take another small example with some math behind artificial network step by step

- i. Feed data to neuron (I assume that a dataset have a value of 4 and I want to calculate prediction)
- ii. Multiply weight with value 4 and send to hidden layer neuron
- iii. Apply sigmoid function to that value and again multiply with some weight and send to output layer neuron
- iv. At last calculated value is our prediction

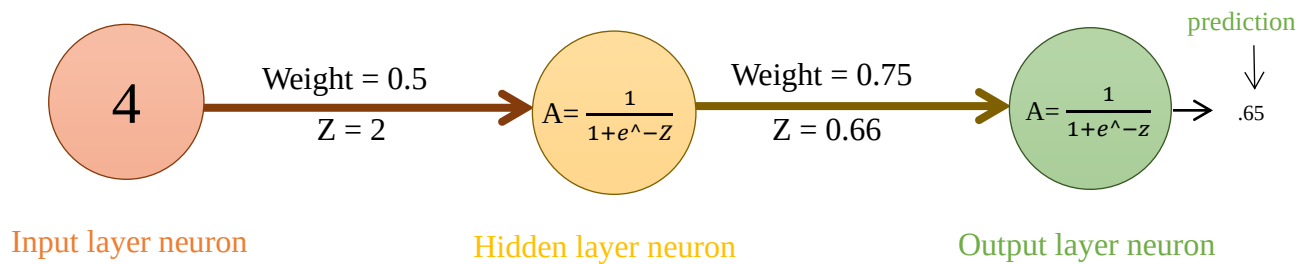


Fig 1.3 Basic working of Neural Network

Weight is automatically selected by neurons it could be any positive number and if the predicted value and actual value have difference (error) then weight automatically get updated backward this is called backpropagation this will continue until the right prediction along with that loss is also reducing.

II. Overview

In this study, I developed an AI to identify COVID-19 patient and risk of having SARS-CoV2 by using their X-ray, CT-scan and some feature.

- **Data mining:**
 - collect useful data of covid patient [10].
 - Extraction useful information from data
 - Create a dataset from unstructured data.
- **Exploratory data analysis:**
 - Feature selection: Identifying major symptoms of COVID-19 patient by analysing their association using different learning approach.
 - Missing data handling: Analysing the correlation between major symptoms, patient age and fill nan value according to that.
 - Encoding: convert categorical feature into 0's and 1's.
- **Model creation:**
 - Machine learning: Developing multiple machines learning model to predict COVID-19 patient's risk.
 - Deep Learning: create convolutional neural network using X-rays and CT-scan.
- **Result and conclusion:** Using statistical analysis to calculate the best fit model according to the study.

III. Research Goals and Approach

The research goal is to achieve a better understanding of COVID-19 with help of Machine learning and Deep learning that help us to build a perfect predictive model and accurate result using AI. This research paper will help us to become familiar with how AI can be useful in COVID-19 pandemic. As you know viruses transmit from one another and it will take 6 to 7 days for symptoms to appear. There are three stages in COVID-19 Low, Moderate, and Critical In the beginning stage it's hard to recognize COVID-19 patient but with the help of AI (Artificial Intelligence), we can predict the outcome probability by using a patient's X-ray, CT-scan, and some common features (symptoms). It's hard to implement practically but it is not impossible. This research is for educational purpose only but it gives a basic understanding of how we can build a better AI that help us in the medical domain.

2. Literature review

Bradly F.Erickson, Panagiotis Korfiatis, Zeynettin Akkus, Timothy L. Kline have conducted research on machine learning in medical domain imaging this author uses classification algorithms like a support vector machine (SVM), K-nearest neighbor (KNN), and neural networks to differ a different disease using images [11].

D.J.S.Sako and J.palimoto from the International Institute of Academic Research and Development (IIARD) done research on classifier system for heart disease diagnosis using only naive Bayesian classifier in this research author extract data from pure raw unstructured data and create data sets and perform analysis but for model training, they use only one algorithm naive Bayes algorithm but accuracy is close enough [12].

V.V. Ramalingam, Ayantan Dandapath, and M Karthik Raja from the department of computer science & engineering, SRM institute of science and technology conducted a research on heart disease prediction using machine learning. They use ensemble technique Algorithm such as support vector machine (SVM), K-Nearest Neighbour (KNN), Naive Bayes, Decision Tree, Random Forest [13].

Sam Royston uses machine learning algorithm to predict the glucose level in blood for diabetes patient. Algorithm used are Ada boost classifier and regressor (boosting algorithm from ensemble technique) and SVM (for classification problem). And the Accuracy is pretty much good [14]

K.M. Al-Aidaroos, A.A. Bakar and Z. Othman have conducted the research for the best medical diagnosis mining technique. For this author compared Naïve Baeyes with five other classifiers i.e., Logistic Regression (LR), KStar (K*), Decision Tree (DT), Neural Network (NN) and a simple rule-based algorithm (ZeroR). For this, 15 real-world medical problems from the UCI machine learning repository (Asuncion and Newman, 2007) were selected for evaluating the performance of all algorithms. In the experiment it was found that NB outperforms the other algorithms in 8 out of 15 data sets so it was concluded that the predictive accuracy results in Naïve Baeyes is better than other techniques.

3. Architecture and design

I. Design strategy

Designing strategy for Machine learning: First, I collected RAW data which is in Chinese then convert it into simple English text, and then moved on the create structured datasets which consist of columns age, gender, cases confirm or not, symptoms, and risk. Secondly, I perform exploratory data analysis (EDA) which consists of analysis, data relation, Data Visualization. Third, encoding: convert categorical feature into 0s and 1s. Fourth, modelling: in modelling, there are multiple algorithms and all algorithms contain two processes training and testing. Fifth, Evaluation testing: evaluate the different model using evaluation matrix and performance matrix and at last result and conclusion

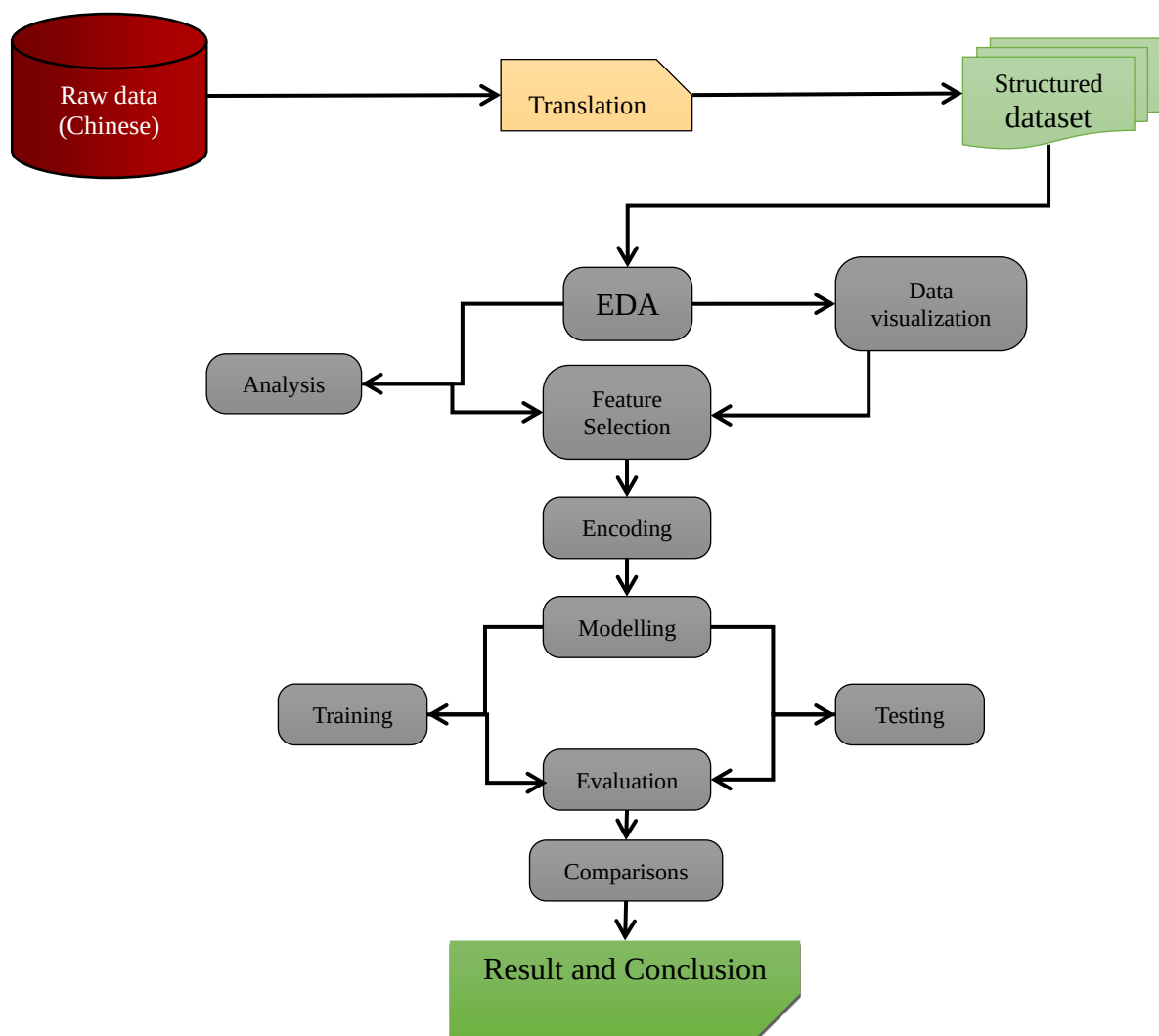


Fig 3.1 Machine Learning Flow Chat

Designing strategy For Deep learning

- Multi-class classification, I collect patient chest X-ray and CT-Scan from various source and then make a class of different disease such as pneumonia virus, pneumonia bacteria, SARS, SARS-CoV2, ARDS, streptococcus, and normal. for multiclass datasets I used a simple

renaming algorithm in visual basic language using metadata file and then selected only particular type of diseases and put it in a class (folder).

- And for binary classification also collect from same source but in this dataset, there are only two class COVID and non-COVID

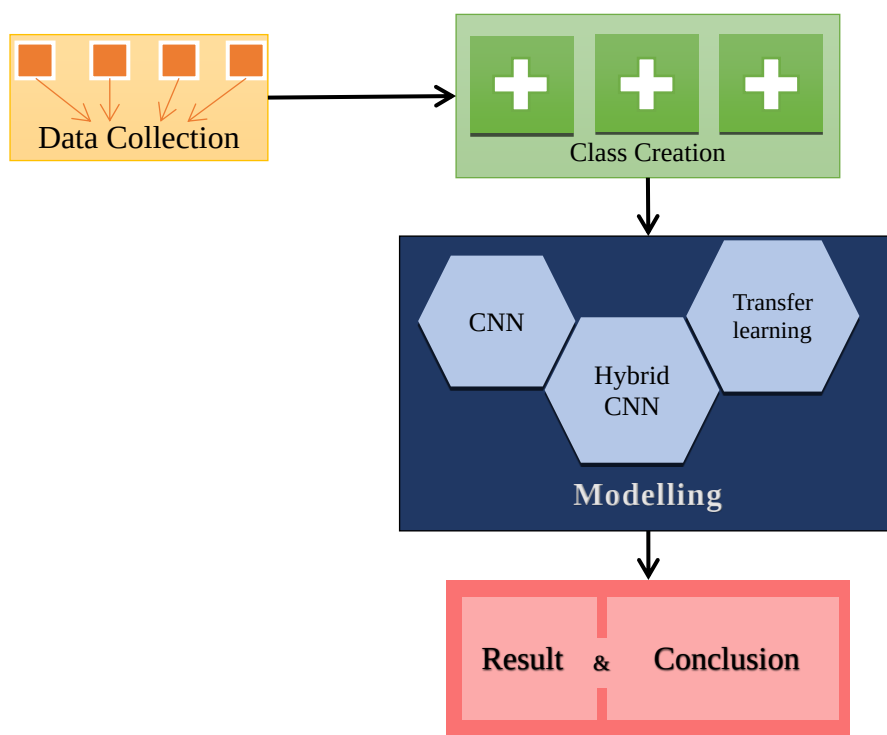


Fig 3.2 Deep Learning Flow Chat

II. Data Analysis

Data analysis helps us to see the structure of data and extract useful information from that. As shown in Figure 3.3 the chat shows covid confirmation cases. According to the analysis, most of the people infected by COVID-19 are between 28-59. This will help to find the missing age value in the dataset.

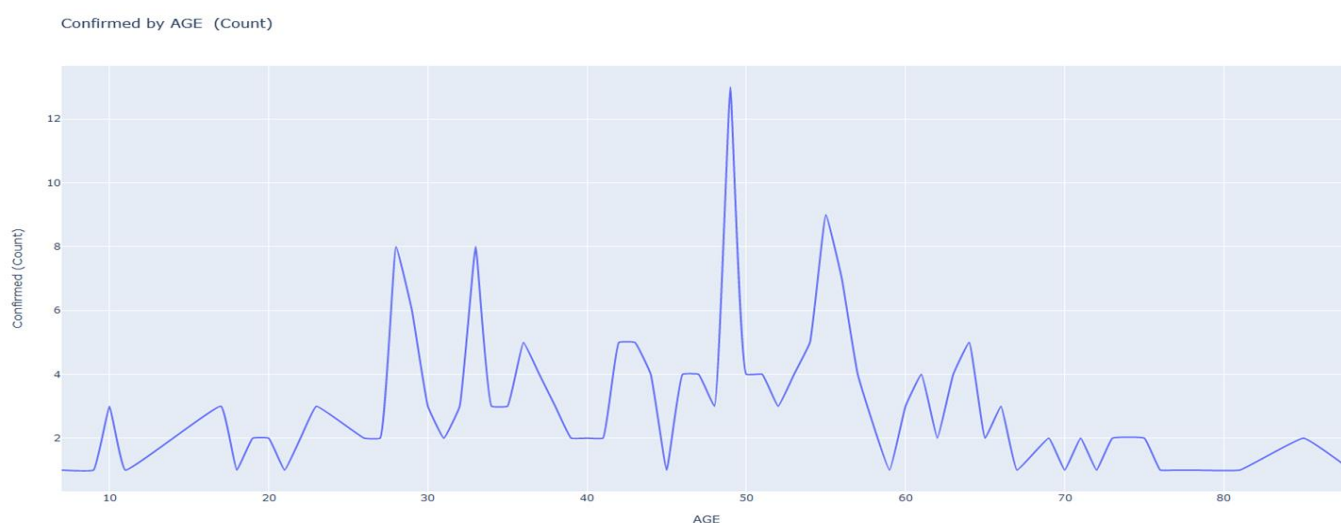


Fig 3.3 COVID Patient Count spike chart

correlation between symptoms. figure 3.4 is analysis of covid symptoms correlation and as you can see fever and dry cough are highly correlated, lung infection and fever are also highly correlated and sore throat correlated with tiredness

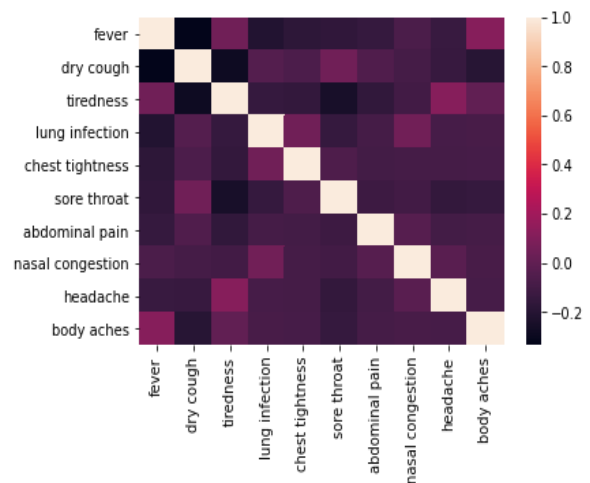


Fig 3.4 symptoms correlation heatmap diagram

III. Parametric Analysis or sensitivity, and uncertainty analysis

Confusion Matrix:

There are some expected errors that may occur while training the machine-like Type-1 and Type-2 error. What is the type-1 and type-2 error? it is an error in which machine predicts the wrong output. Let's have look at the confusion matrix [15].

Predicted value	Actual value	
	TRUE	FALSE
	Positive	Negative
Positive	TP	FP
Negative	TN	FN

Table 1 Confusion Matrix

above matrix shows that if a person is not infected with COVID-19 but the machine predicted true then it is known as type-1 error and if a person is infected and the machine predicted false then it is known as type-2 error usually type-2 error is more dangerous. To overcome this problem, I used pure and real-life data and also try to make an optimized model.

Precision and Recall:

Precision and recall are used for information retrieval example when we search something in google there are millions of records based on the result but the topmost record is relevant to search result this is due to precision and recall.

$$precision = \frac{True\ positive}{True\ positive + False\ positive}$$

The precision formula shows that Out of the total predicted positive result by the model what is the percentage of the actual result. Same goes for recall when we want to measure our model true positive rate, we use recall

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

F1-score: It is nothing but a harmonic mean of precision and recall. Rather than balance precision and recall individual we can also direct look for F1-score.

$$F1\ score = 2 * \frac{precesion * recall}{precesion + recall}$$

Bias and Variance [16] [17]:

In supervised machine learning whenever we study generalize models it is compulsory to know prediction errors (Bias and Variance). Bias and variance are a concept that helps us to minimize the errors and avoid mistakes like underfitting and overfitting.

What Is Bias?

In simpler terms, bias is a difference between actual training data points and a prediction. A model with high bias always leads to a less accurate for both training and testing. Our goal is to create models with low bias.

What Is Variance?

Variance is an error while performing testing because the model is too much fit with training data and does not generalize on data that it hasn't seen before this is known as high variance. As a result, this type of model performs very well on training but has high errors in testing.

Let's study what problems created with high bias and high variance.

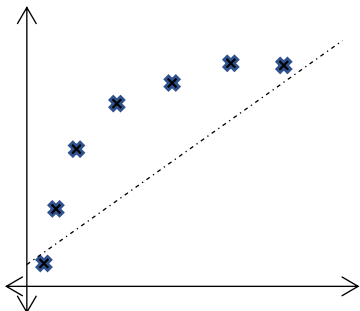
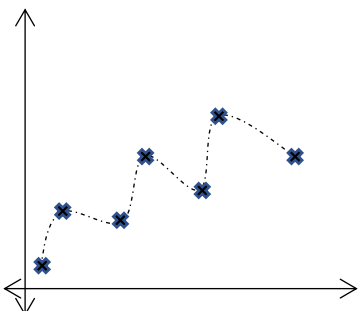
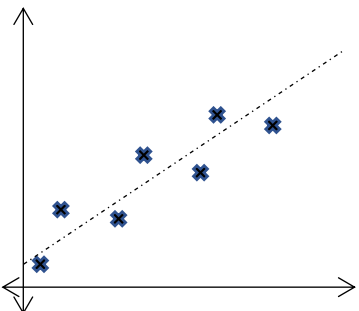
Underfitting example	Overfitting example	Generalize example
		
Underfitting is a problem when a model has high bias above diagram indicated that best line not even fit on a training data.	Overfitting is a problem when model it too good with training data but cannot predict unseen data or test data as shown in above diagram.	This is a generalize model where our best fit line is not underfitting neither overfitting.
Training and Testing both are imperfect	There is a very huge difference in training and testing accuracy. Training is good but testing is inferior	In generalize model there is a very less difference in training and testing accuracy

Table 2 bias and variance examples

For achieving good accuracy and lower error rate our model should be in low bias and low variance.

4. Methodology and proposed solution

I. Modelling

a. Algorithms

(1) Decision Tree: A decision tree is a flowchart-like structure it consists of three components root node leaf node and branches. It is usually used for classification problem. The decision tree takes the decision on probability or possibility like for example if I have a fruit basket and I want to separate apple from basket first decision tree run a condition diameter > 7 cm if true then go to next possible node then again run a condition colour = red If true then separate and if not then go on.

In order to split a decision tree from root node to leaf node there is a concept Entropy [18]. It means purity of the subsets or you can also say that measure of purity. The mathematical formula for entropy is $E(s) = \sum_{i=1}^c -p_i \log_2 p_i$ where p_i stands for probability of element. For example, we have a binary classification problem and our dataset consist of 100 records. 30 records belong to positive and 70 belongs to negative class. then probability of positive class will be $+p = \frac{30}{100}$ and $-p = \frac{70}{100}$ for negative class.

Let's put this in equation

$$E(s) = -\frac{3}{10} * \log_2 \left(\frac{3}{10} \right) - \frac{7}{10} * \log_2 \left(\frac{7}{10} \right) \approx 0.88$$

The entropy is 0.88 which is very good or datasets have very less impurity. Entropy measure in between 0 to 1. Entropy also depends on number of classes in dataset.

(2) Random Forest [19] [20]: This is from the ensemble technique (Ensemble learning is a machine learning paradigm where multiple models are trained to solve the same problem and combined to get better results) and also called a bootstrap aggregator or bagging usually uses for classification and regression type problems. Random forests consist of many decision trees. A decision tree is like a flowchart spreading from top to bottom. Basically, Random forest uses the vote of all decision tree and give the most frequent outcome. As shown in figure 4.2 first dataset divides into 3 decision trees it may n number of trees depending on the problem. Then decision tree predicts the outcome and the last most frequent prediction will become random forest prediction. Random forest solves the problem of low bias and high variance in the decision tree. First, understand low bias and high variance. Basically, Low Bias means a low error in training and high variance means a high error in testing. In the decision tree, if we train a machine to its complete depth then it has a low bias but whenever a new test data came it will show a high variance also called overfitting. In the random forest, I am using multiple

decision trees and we know that each and every decision tree have high variance but when we combine all the decision tree using bootstrap aggregator with respect to majority vote then for every decision tree in random forest high variance converted into low variance. The random forest also uses

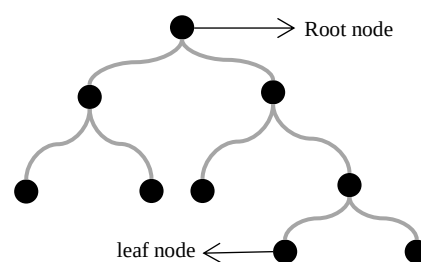


Fig 4.1 decision Tree

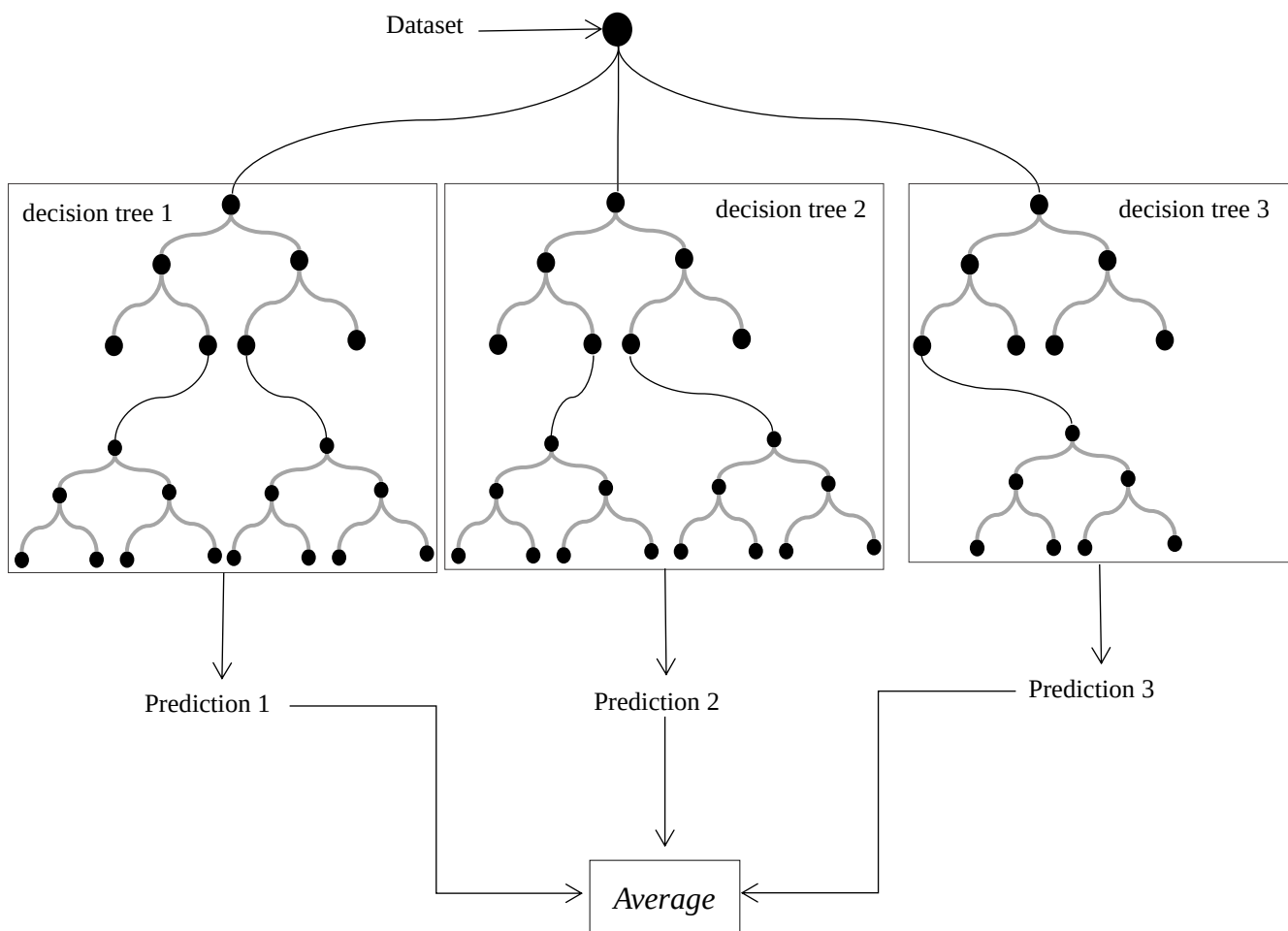


Fig 4.2 Random Forest

as a regressor but for regression problem, the output will be the mean of all decision trees because all the decision trees return the continuous value.

(3) Adaptive Boost (Ada Boost) [21]: Ada boost is from boosting algorithm family. Ada boost also uses multiple decision trees or it may be random forest but the idea behind Ada boost is it will train a model and pass that model error to another model with updated sample weight

($W = \frac{1}{\text{number of record}}$ or $\frac{1}{n}$) and the cycle continues to the last decision tree. Let's understand with an example suppose I have a dataset with three features and it is a binary classification problem. What Ada boost does is create individual decision trees or stumps for all feature in the dataset and this is a sequential model so the first stump is selected on the basis of entropy. If the entropy of a particular stump is low then it will use as a first stump. Second, it will calculate the total error of the first stump by adding all error and the third performance of a stump is calculated it is denoted by

$$\text{performance}(p) = \frac{1}{2} \log_e \left(\frac{1 - TE}{TE} \right)$$

↑
Total Errors

TE is nothing but total error. After a performance of a stump calculated then only error passes to another stump and this will continue till the end. Last update the weight of incorrectly classify there is a simple formula $c = \text{old weight} * e^p$ and $c1 = \text{old weight} * e^{-p}$ for correctly classify weight.

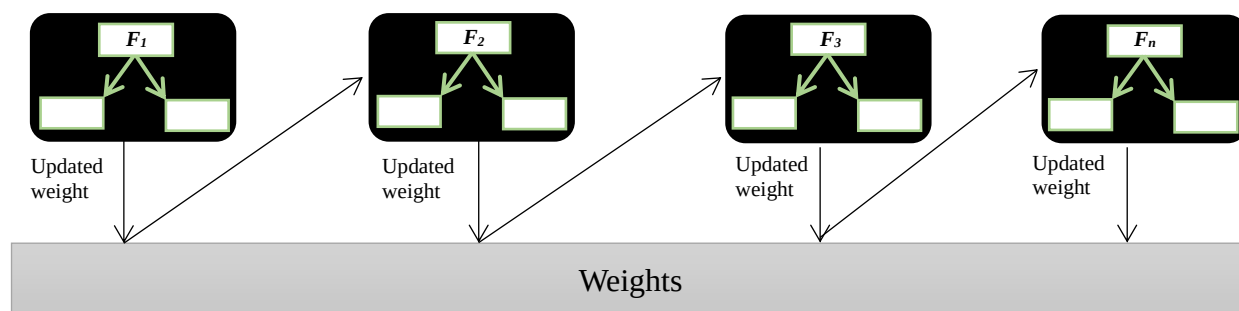


Fig 4.3 Adaptive boost

The above figure shows how the weight gets updated each and every time for a new stump (decision tree) until the last one. This boosting technique allow us to focus on mainly errors and boost the accuracy.

(4) Logistic Regression [22] [23]: It is a statistical model that is only used for binary classification. Logistic regression also uses sigmoid function what logistic regression do is predict the class of dependent categorical feature on the basis of an independent feature.

In figure 4.4 there are some data points on a curve line called as sigmoid curve. On Y-axis there are two classes 1 and 0 and, on the X-axis, there is a continuous variable. The sigmoid curve uses continuous values to predict binary classes. On the sigmoid curve, there is a mid-value called a threshold value. It is used when a certain value is higher than 0 but less than 1 or vice-versa. Like for example if the value is greater than 0.5 then it considers as 1 and below 0.5 consider as 0

In this research I used logistic regression for multi-classification for that there is a simple technique called one versus rest. In OVR model only consider two classes. If class 1 is

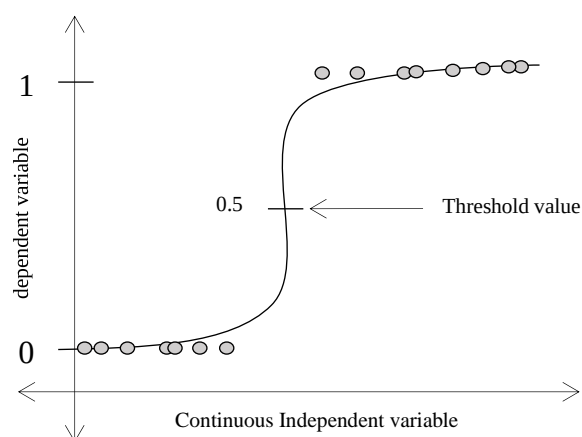


Fig 4.4 Logistic regression

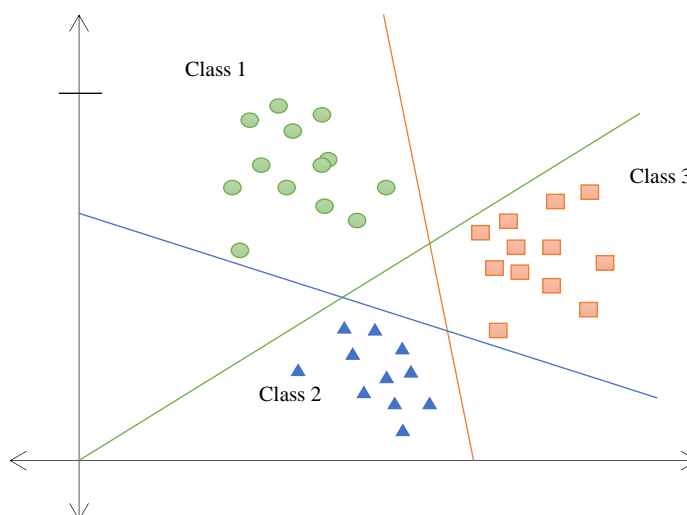


Fig 4.4 multi class Logistic regression

selected then all other classes become one class. multiclass logistic regression model train on iteration. Fig 4.4 shows that in the first iteration class 1 is selected and the other two classes are treated as one class. In the second iteration class 2 is selected and the other two classes are treated as one class. and last class 3 is selected and the other two classes are treated as one class. After that, it will generate three different outputs based on their probability. The highest probability of a class considers as the final output. There are two different ways for class prediction, first cost function, and second gradient descent. Cost function gives an idea of how far our prediction from the original output here is a formula for a cost function.

$$C = -\frac{1}{n} [\sum y^i \log h^i + (1 - y) \log(1 - h^i)]$$

Original output
 Number of training data
 Predicted outcome

And another way is using gradient decent using this formula we can also calculate loss between actual and predicted value

$$\theta = \theta - \alpha \sum (h^i - y^i) X_j^i$$

Value of theta update on each iteration.

(5) SVM (Support Vector Machine) [24] [25] [26]: SVM is another simple algorithm that every machine learning expert should know. SVM is one of the best algorithms for classification and regression but it is widely used in classification. Let's understand the working of SVM the objective of SVM is to find the best hyperplane in between different classes and create a marginal distance from hyperplane to a certain distance and point that are on margin or inside the margin are called support vector because those points help a machine to calculate distance from hyperplane so that whenever a new data point comes then the machine can easily classify from what class it belongs to.

Figure 4.5 illustrates how two classes are separated by a linear line called a hyperplane because in a 2D structure plane looks like a line but in the 3D figure, we cannot separate data point by-line we need a plane that's why it is called a hyperplane this mean SVM also work with 3D or infinite dimension problems. Left and right side of a hyperplane there is a dashed line called a margin and those points that are on the margin are support vectors. In SVM we are looking for a large marginal distance for easy and accurate classification.

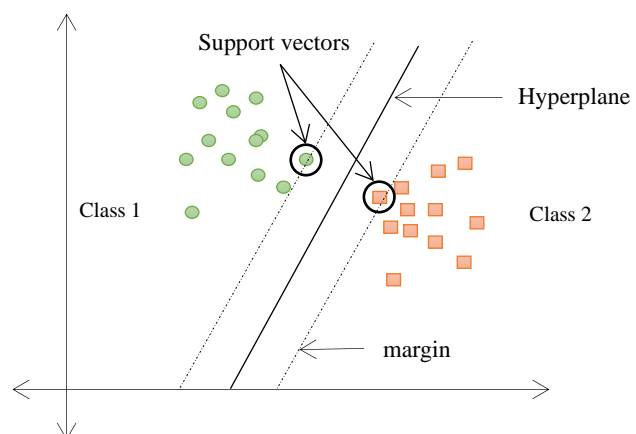


Fig 4.5 support vector classifier

Diagram illustrating the components of Bayes' Theorem:

- Hypothesis**: Points to $P(H|E)$
- Given that**: Points to $P(E|H)$
- Probability**: Points to $P(H)$
- Evidence**: Points to $P(E)$

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Bayes theorem says that probability of a hypothesis (A proposition made as a basis for reasoning, without any assumption of its truth) given that on evidence. It means the probability of a hypothesis is true based on a shred of given evidence or $P(H|E)$. we are restricting our view only to the possibilities where the evidence holds. Were $P(E|H) * P(H)$ says that a probability of hypothesis based on a true hypothesis multiply by a probability of a total number of the true hypothesis and $P(E)$ it is a total no of a probability of hypothesis based on a true hypothesis multiply by a probability of a total number of the true hypothesis and a total no of false hypothesis multiply by evidence based on a false hypothesis basically

$$P(E) = P(E|H) * P(H) + P(E|\neg H) * P(\neg H)$$

Scientists use Bayes theorem to when analysing the extent to which new data validates or invalidates their model. And programmer use bayes theorem in A.I.

Let's take an example how naïve bayes classifier work suppose we have a datasets $x = \{X_1, X_2, X_3, X_4, X_5, \dots, X_n\}$ $\{Y\}$ here X is a independent feature and Y is a dependent feature or outcome.

Put this data set in bayes theorem formula

$$P(Y|X_1, X_2, X_3, X_4, X_5, \dots, X_n) = \frac{P(X_1|Y).P(X_2|Y).P(X_3|Y).P(X_n|Y) * P(Y)}{P(X_1).P(X_2).P(X_3).P(X_4) \dots \dots P(X_n)}$$

also represent as

$$P(Y|X_1, X_2, X_3, X_4, X_5, \dots, X_n) = \frac{P(Y) * \pi_{i=1}^n P(X_i|Y)}{P(X_1).P(X_2).P(X_3).P(X_4) \dots \dots P(X_n)}$$

Denominator is constant so we remove and the remaining equation is directly proportional to each other

$$P(Y|X_1, X_2, X_3, X_4, X_5, \dots, X_n) \propto P(Y) * \pi_{i=1}^n P(X_i|Y)$$

And then find argmax. Argmax is an operation that find the argument that gives the maximum value for a target function

$$Y = \operatorname{argmax}_Y P(Y) * \pi_{i=1}^n P(X_i|Y)$$

This our final outcome for given dataset

There are three different types of naïve bayes

1. Multinomial naïve Bayes: mostly used in document classification problem example: - whether a document belongs to a category of formal, politics, sports, technology, etc. multinomial naïve Bayes use frequency of a word present in a document for classification.

2. Bernoulli naïve Bayes: This predictor used for binary classification like yes or no, true or false, 1 and 0, etc.
3. Gaussian naïve Bayes: This predictor used when the problem is continuous and is not discrete. It uses gaussian normal distribution. Therefore, the way the value of the dataset changes, the formula also changes.

$$P(X_i|Y) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{(X_i - \mu_Y)^2}{2\sigma_Y^2}\right)$$

Naïve Bayes classifier is a very useful predictor it is mainly used in sentiment analysis like spam detection, filtering, recommendation, etc but the problem is requirements of a predictor to be independent which is not always in real-world data, this hinders the performance of the classifier

(7) Hybrid CNN (Convolutional Neural Network): Before going on hybrid CNN, we need to understand CNN [30]. It is an image classification algorithm. Convolutional neural networks inspired by the visual cortex. The visual cortex is a posterior part of the brain that is sensitive to specific regions of the visual fields. Some neuron cells in brain are active only in presence of the edge of a certain orientation. For example, some neuron cell only fires on a vertical edge or some other fires on the horizontal edge.

Why we need a convolutional neural network we already have ANN (Artificial Neural Network)?

Because an artificial neural network uses a fully connected layer means each hidden layer neuron is connected to the all-input neurons which is a bad idea. If we have an image size 200 x 200 x 3 pixels in a fully connected layer then the number of weights required in the first hidden layer will be 120,000. In a convolutional neural network, a neuron in a layer is only connected to a small region of the layer this will help us to handle less amount of weight and also less neurons to handle.

Let's understand how Convolutional Neural Networks works. It consists of four layers

1. **Convolutional:** In the convolutional layer, we will move the filter to every possible position on the image and multiply each image pixel by the corresponding filter pixel (filter/feature) and add them to form a new output also called a feature. There are many filters we can set according to our need.

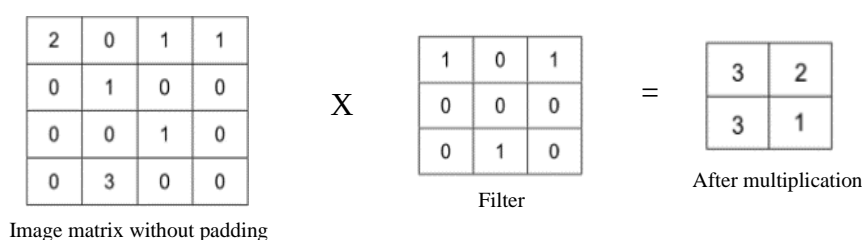


Fig 4.6 filter multiplication with image matrix

- 2. ReLU (Rectified Linear Unit) Layer:** ReLU is nothing but an activation function. ReLU is an activation function that only activates neurons if the value is greater than or equal to zero and remains zero if the value is less than zero but when the input rises above a certain threshold it has a linear relationship with the dependent variable as shown in figure 4.6. ReLU layer also removes all the negative value and replace with zero from the matrix.

2	1
-1	0

For the above matrix ReLU activation function activate except the -1 because it is less than zero.

After the convolutional layer, the ReLU layer is responsible for when to activate a neuron according to the filter

Formula of ReLU activation function is $\max(Y, 0)$ where

$$Y = \sum_{i=1}^n w_i x_i + b_i$$

W_i = weight

X_i = input feature

b_i = bias

suppose Y is negative then the formula is $-\max(-Y, 0)$, and the output is always zero

We only use the ReLU activation function in Convolution Neural Network because the ReLU function has a linear relationship with the dependent variable.

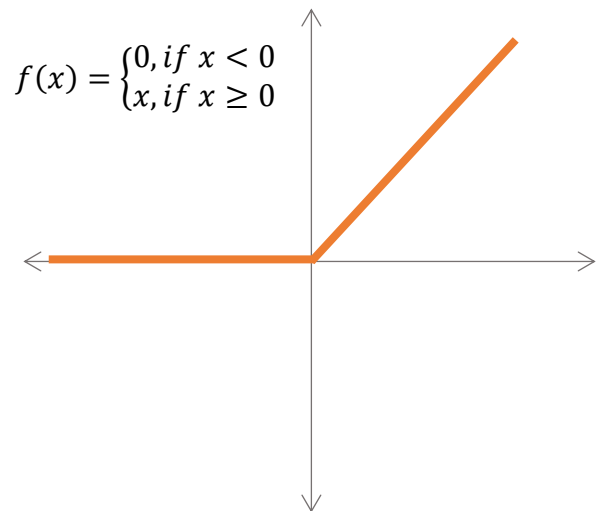


Fig 4.7 ReLU activation function

- 3. Pooling Layer:** pooling layer is used to reduce the size of an image width and height and depth is determined by a colour channel. What pooling layer does is pick the maximum value from a certain size of a matrix and make a new small matrix there are two type of pooling max-pooling and average pooling as shown below in fig 4.8

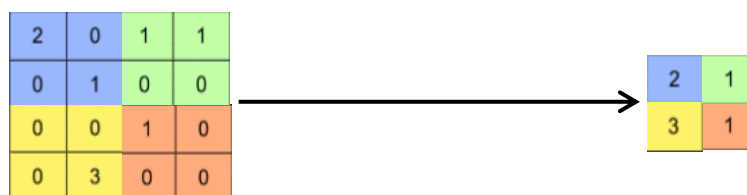


Fig 4.8 max-pooling

- 4. Fully connected layer:** After convolutional, ReLU and Pooling layer our image matrix is shrink down and last, we need to flatten the shirked matrix into a single stack and pass it to fully connected layer in fully connected layer there is another activation function that helps to classify an image.

This is how convolutional neural works. hybrid Convolutional Neural Network is a combination of CNN with Machine learning classification algorithm. In this research I used CNN with SVM both are classification algorithm.

(8) VGG16 (Visual Geometry Group) [31]: VGG-16 was proposed by Karen Simonyan and Andrew Zisserman in Oxford university on 2014. VGG16 is a predefined Convolutional Neural Networks whose weight are already defined. VGG-16 uses image net API which contain 14 million images, and 1000's of classes. Image net dataset contain fixed sized images (224 x 224 x 3) as input. VGG 16 contain 13 convolutional layers with ReLU activation function, 5 pooling layer and 3 dense fully connected layer with activation function and loss depending on a classification type.

b. Loss Function

The loss function is one of the important topics in data science. Loss function helps us to understand that how far our prediction from the actual value and reduce the loss as much as can to gain higher accuracy with the help of an optimizer. There are many different loss functions but I am talking about only loss functions that are used in this research.

- Cross entropy [32] or log loss, logistic loss formula for cross entropy is

$$CE = - \sum_{i=1}^n T_i \log(P_i), \text{ for } n \text{ classes}$$

Were T_i is ground truth label and P_i is the probability of the classes

- Binary cross-entropy [32] [33]: It is a sigmoid activation function plus a cross-entropy loss. Binary cross entropy only calculates the errors of two classes from ground truth also called sigmoid loss.

formula for binary cross entropy is

$$C = -\frac{1}{n} \sum_x [y \log a + (1 - y) \log(1 - a)]$$

y = Desired output or actual output

x = input

$z = wx + b$ (weight * input + bias)

$a = \sigma(z)$ (sigmoid of z . It will always in between 0's and 1's but not 0 and 1)

If $y = 0$ then our equation become

$$C = -\frac{1}{n} \sum_x [\log(1 - a)]$$

because “ $\log a$ ” is multiply by “ y ” which is zero so whole equation depend on “ a ” and it should be very close to zero.

And if $y = 1$ then our equation is

$$C = -\frac{1}{n} \sum_x [y \log a + \log(1 - a)]$$

- Categorical cross-entropy: It is a SoftMax activation function plus a cross-entropy loss also called SoftMax loss. Categorical cross-entropy is used to calculate the errors of multiple classes from the ground truth. The main difference between binary cross-entropy and categorical cross-entropy is the activation function. SoftMax function compressed the output probability of all the classes in between 0's to 1's so when we add the total probability of classes it will always be 1. If the probability of one class goes high then all other classes' probability goes down.

SoftMax function formula is

$$s(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

Were,

e^{y_i} = standard exponential function for input vector

e^{y_j} = standard exponential function for output vector

- Hinge Loss [34] [35]: It is special type of loss function used for maximize the margin, most notably for support vector machines (SVMs). Formula of hinge loss

$$H_l = \max(0, 1 - y^i \hat{z})$$

Were,

y^i = correct label

\hat{z} = how far from decision boundary

c. Optimizer

Optimizer is algorithm or method used to change learning rate and weight of neural network to reduce loss function.

Optimizer used in research is Gradient Descent and ADAM optimizer.

- Gradient Descent optimizer: After prediction and calculating loss function our aim to reduce loss function by using gradient decent. what gradient does is update the weight of all neurons in backpropagation.

Formula of Gradient descent is

$$W_n = W_o - \eta * \frac{d_l}{d_{W_o}}$$

Were,

W_n = new weight,

W_o = old weight,

η = lerning rate (steps taken to reach global minima),

$\frac{d_l}{d_{W_o}}$ = derivatives of loss with respect to old weight.

- ADAM optimizer [36]: ADAM optimizer is one of the best optimizers also known as Adaptive moment estimation. It required less memory and very efficient. ADAM optimizer is a combination of gradient descent, momentum and RMSP (Root Mean Square Propagation)

Momentum: It is used to minimize error and reach global minima of gradient descent by creating force or motion to overcome local minima. As shown in figure 4.9 optimizer should be in global minima or less error for that it needs to be cross little hump this is done by momentum.

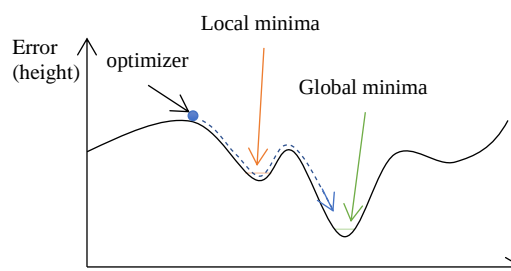


Fig 4.9 working of optimizer

RMSP (Root Mean Square Propagation) [37]:

RMS prop is like gradient descent with

momentum but the huge difference is RMS prop restrict the vertical oscillation therefore we can increase learning rate and it could take larger step toward global minima.

II. Performance Evaluations

For the Machine learning model, I selected 12 features (age, gender, fever, dry cough, tiredness, lung infection, chest tightness, sore throat, abdominal pain, nasal congestion, headache, body aches) from raw data and created a new dataset. The new dataset consists of 200 genuine covid patience. Patience raw data collected from (<https://github.com/BDBC-KG-NLP/COVID-19-tracker.git>). The feature is also known as the dependent variable. The dependent variable helps us to predict the risks of having covid-19 based on the explicit features. The risk column in the dataset has three classes high, moderate and low. First, I divide the whole dataset into four-part x_{train} , y_{train} , x_{test} , and y_{test} . Where x is the independent variable and y is the dependent variable x_{train} data and y_{train} data is used for machines to understand the relationship between an independent variable and dependent variable. x_{test} and y_{test} are used for testing machine accuracy based on the new data. Second, I pre-process the data and make it in a standard scalar. Standard scalar is nothing but a standard normal distribution what standard normal distribution does is compress the one-hot encoded data (basically 0's and 1's) into -1 to 1 or make data into zero centered data. Third, feed data to various machine learning algorithms. And last evaluate all the model using confusion matrix, classification matrix, and training, testing score.

For deep learning, I collect data from various sources

(https://data.mendeley.com/datasets/8h65ywd2jr/3#_sid=js0,

<https://www.kaggle.com/bachrr/covid-chest-xray?select=metadata.csv>) there are approx. 26,700 images of covid-19, ARDS, pneumonia bacteria, pneumonia virus, SARS, streptococcus's patience chest x-ray, and CT scan all the images are totally random and mix. first, I need to take classes on different diseases. along with data, I got a metadata.csv file also that contain all the images label and diseases so by using that metadata file I replaced images original label with diseases name for all the 26,700 images by using a simple rename code on a visual basis after that search, a particular diseases name in the data folder and all the same name file appear then select that all similar images and put into another folder, and rename that folder with particulate diseases name and that is one class, I followed this procedure until all the images are sorted then I make another folder with two classes one for covid patience chest x-ray and CT-scan and another for non-covid x-ray and CT-scan. whole efforts are only for creating multiclass classification data that contain seven different diseases x-ray and CT-scan and another one is for binary classification data that contain two classes covid and non-covid images.

VGG16 is a predefined CNN model built-in Keras library and as I mention above it's only accepted 244 x 244 size images so I converted all the images in that size and the last output layer of VGG16 contain 1000's neurons or 1000's of classes so I deleted the last layer and then edit to 7 classes or 7 layers for multiclassification and for binary I use only one neuron. Last I loaded weight from ImageNet API

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 1)	25089
Total params: 14,739,777		
Trainable params: 25,089		
Non-trainable params: 14,714,688		

Fig 4.10 VGG 16 architecture

Fig 4.10 is VGG 16 architecture edited according to research need and requirements It is a binary classification architecture.


```

Model: "sequential"
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 32, 32, 32)         896
max_pooling2d (MaxPooling2D) (None, 16, 16, 32)         0
conv2d_1 (Conv2D)            (None, 16, 16, 32)         9248
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 32)         0
flatten (Flatten)            (None, 2048)                0
dense (Dense)                (None, 128)                262272
dense_1 (Dense)              (None, 7)                   903
-----
Total params: 273,319
Trainable params: 273,319
Non-trainable params: 0

```

Fig 4.11 Convolutional Neural Network Architecture

The above image is an architecture of CNN used in this research for image classification first, I used convolutional 2D layer or conv2D and ReLU layer and as you can see output shape of the image is changed by 32 x 32. it is the size of the image and another 32 is a filter depth second, I used max-pooling layer to scale down an image but after all this process our CNN still needs 8,192 neurons so I used again conv2d, ReLU, and pooling layer to minimize the neurons after this I get 8 x 8 size image which gets multiply by 32 filter which is nothing but a total no of neurons 2048 after this, I flatten an image matrix into a single stacked and last fully connected layer. In a fully connected layer, there are two-layer. One is the hidden layer and the other one is the output layer. In the hidden layer, I used 128 neurons and the output layer has only 7 neurons which are based on the number of classes. Afterward I created CNN model

<i>MODEL</i>	<i>Classification Type</i>	<i>Activation Function</i>	<i>Loss Function</i>	<i>Regularization</i>
<i>CNN</i>	Binary	ReLU and Sigmoid	Binary-Cross Entropy	-
	Multiclass	ReLU and Softmax	Categorical-Cross Entropy	-
<i>Hybrid CNN</i>	Binary	ReLU and Linear	Hinge	L2 norm
	Multiclass	ReLU and Softmax	Squared Hinge	L2 norm
<i>VGG 16</i>	Binary	ReLU and Sigmoid	Binary-Cross Entropy	-
	Multiclass	ReLU and Softmax	Categorical-Cross Entropy	-

Table 3 deep learning model basic info

5. Validation of modelling and Result

In the below table, there are various machine learning algorithms used to achieved best accuracy score.

<i>ALGORITHMS</i>	<i>Precision</i>		<i>Recall</i>		<i>F1-Score</i>		<i>Training accuracy</i>	<i>Testing accuracy</i>
<i>Decision Tree</i>	High	0.88	High	0.33	High	0.48	97%	66%
	Low	0.50	Low	0.62	Low	0.55		
	Moderate	0.14	Moderate	0.80	Moderate	0.24		
<i>Optimized Decision Tree</i>	High	0.62	High	0.91	High	0.74	95%	72%
	Low	0.81	Low	0.62	Low	0.70		
	Moderate	0.71	Moderate	0.71	Moderate	0.71		
<i>Random Forest</i>	High	0.62	High	0.37	High	0.47	94%	83%
	Low	0.31	Low	0.62	Low	0.42		
	Moderate	0.46	Moderate	0.52	Moderate	0.49		
<i>Support Vector Machine</i>	High	0.81	High	1.00	High	0.90	84%	83%
	Low	0.94	Low	0.71	Low	0.81		
	Moderate	0.79	Moderate	0.85	Moderate	0.81		
<i>Naïve Bayes</i>	High	1.00	High	0.81	High	0.90	82%	83%
	Low	0.71	Low	0.94	Low	0.81		
	Moderate	0.85	Moderate	0.79	Moderate	0.81		
<i>K-Nearest Neighbour</i>	High	1.00	High	0.62	High	0.77	81%	77%
	Low	0.70	Low	0.88	Low	0.78		
	Moderate	0.73	Moderate	0.79	Moderate	0.76		
<i>Adaptive Boost</i>	High	0.90	High	0.56	High	0.69	97%	70%
	Low	0.59	Low	0.81	Low	0.68		
	Moderate	0.71	Moderate	0.71	Moderate	0.71		
<i>Voting Classifier</i>	High	1.00	High	0.81	High	0.90	87%	85%
	Low	0.75	Low	0.94	Low	0.83		
	Moderate	0.85	Moderate	0.82	Moderate	0.84		

Table 4 Algorithms performance table

Below table is a deep learning model performance table

<i>DL models</i>	<i>Class type</i>	<i>Training loss</i>	<i>Training accuracy</i>	<i>Testing loss</i>	<i>Testing accuracy</i>	<i>Total accuracy</i>
<i>CNN</i>	binary	0.12	0.94	0.31	0.80	81%
	multiclass	0.43	0.81	0.43	0.88	87%
<i>Hybrid CNN</i>	binary	0.20	0.91	0.47	0.82	80%
	multiclass	0.97	0.79	0.93	0.86	85%
<i>VGG 16</i>	binary	0.15	0.89	0.50	0.79	80%
	multiclass	0.29	0.87	0.44	0.88	87%
<i>Optimized CNN</i>	multiclass	0.34	0.85	0.31	0.90	90%

Table 5 DL models result

6. Conclusion

By analyzing experimental research, I found that it's hard to recognize coronavirus sometimes covid test also failed we don't have a proper test kit to check COVID-19, but chest x-ray and CT-scan can help us to find coronavirus but there is also confusion because of pneumonia and coronavirus makes same symptoms on the chest and it's hard to find coronavirus in beginning stages also but only for human's, so I decided to conduct research that makes an idea to find COVID-19 more accurately and I ended up with A.I. I found that Artificial Intelligence not only useful in the techie world but also in the medical domain. And last I completed my research and here is my conclusion and result.

Machine learning and deep learning model that are used in this research they all are performing well in predicting diseases I made two types of predicting analysis first explicit prediction in this I used symptoms that easily identified or explicit changes occur in diseases like weakness, nasal conjunction, body pain, etc. I ran an analysis on various machine learning classification algorithms that predicts how much a person has a risk of COVID-19 based on their explicit symptoms and in that analysis SVM, Naïve Bayes classifier and voting classifier is given the best performance result but this is not enough because the machine uses only explicit symptoms data so I decided to use convolutional neural networks which uses patience chest x-rays and CT-scan and found patience is infected or not. I made multiple CNN and Hybrid CNN models to achieve what I want to accomplish. In this analysis, CNN not only predicts accurate diseases but also predicts even in the beginning stages. All CNN, hybrid CNN and transfer learning given the best result but I need more accuracy because in the medical domain one single mistake can cost a life so I make a new optimized CNN which is best and given accuracy $\approx 90\%$ and loss reduced to $\approx 32\%$.

This research is for educational purpose only

7. References

- [1] A. F. S. a. M. C. Hawn, "An Apparently New Respiratory Disease of Baby Chicks," *American Veterinary Medical Association*, vol. 78, pp. 413-422, 1931.
- [2] C. B. L.D. Bushnell, "Laryngotracheitis in Chicks," *Poultry Science*, vol. 12, no. 1, pp. 55-60, 1933.
- [3] C. D. H. a. F. R. Beaudette, "Cultivation of the virus of infectious bronchitis," *American Veterinary Medical Association*, vol. 90, p. 51-60, 1937.
- [4] J. S. M. P. Kahn and K. M. McIntosh, "History and Recent Advances in Coronavirus Discovery," *The Pediatric Infectious Disease Journal*, vol. 24, no. 11, pp. S223-S227, November 2005.
- [5] D. G. (. Decaro N. Tidona C, "The Springer Index of Viruses," *Springer. Alphacoronavirus*, p. 385-40, 2011.
- [6] J. McCarthy, "The History of Artificial Intelligence," no. December 2006, 1956.
- [7] I. S. N. B. R. P. D. D. Parth Bhavsar, "Machine Learning in Transportation Data Analytics," *Data Analytics for Intelligent Transportation Systems*, no. 2017, pp. 283-307, 1959.
- [8] Rina Dechter, "Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition," *Artificial Intelligence*, vol. 41, no. 3, pp. 273-312, 1990.
- [9] "Brain Tumor Center," Johns Hopkins Medicine, [Online]. Available: [https://www.hopkinsmedicine.org/neurology_neurosurgery/centers_clinics/brain_tumor/about-brain-tumors/how-the-brain-works.html#:~:text=The%20cerebrum%2C%20the%20large%2C%20outer,halves\)%3A%20left%20and%20right.](https://www.hopkinsmedicine.org/neurology_neurosurgery/centers_clinics/brain_tumor/about-brain-tumors/how-the-brain-works.html#:~:text=The%20cerebrum%2C%20the%20large%2C%20outer,halves)%3A%20left%20and%20right.)
- [10] [Online]. Available: <https://github.com/BDBC-KG-NLP/COVID-19-tracker.git>,).
- [11] Bradley J., Erickson, , Panagiotis Korfiatis; , Zeynettin Akkus,; , Timothy L. Kline,, "Machine Learning for Medical," *informatics*, p. 505 to 515, 2017.
- [12] D. J. S. S. & J. Palimote, "A Medical Document Classification System for Heart Disease," *Diagnosis Using Naïve Bayesian Classifier*, vol. 4, p. 79, 2018.
- [13] A. D. M. K. R. V.V. Ramalingam*, "Heart disease prediction using machine learning techniques," vol. 7, pp. 684-687, 2018.
- [14] S. Royston, *Practical Machine Learning for Diabetes Care*, no. December 16, p. 8, 2014.
- [15] r. agrawal, "The 5 Classification Evaluation metrics every Data Scientist must know," 17 September 2019. [Online]. Available: <https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226>.
- [16] s. sing, "Understanding the Bias-Variance Tradeoff," towards data science, 21 may 2018. [Online]. Available: <https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229>.
- [17] K. Naik, Director, *Machine Learning-Bias and Variance in Depth Intuition/ Overfitting Underfitting*. [Film]. INDIA: Krish naik, 2020.
- [18] S. T, "Entropy: How Decision Trees Make Decisions," 11 January 2019. [Online]. Available: <https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8>.
- [19] t. yiu, "Understanding Random Forest," 12 June 2019. [Online]. Available: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>.
- [20] K. Naik, Director, *Random Forest Classifier and Regressor*. [Film]. INDIA.2019.
- [21] k. naik, Director, *adaptive boost*. [Film]. India .2019.
- [22] R. n. Sucky, "Multiclass Classification Using Logistic Regression from Scratch in Python: Step by Step Guide," towardsdatascience.com, 5 September 2020. [Online]. Available: <https://towardsdatascience.com/multiclass-classification-algorithm-from-scratch-with-a-project-in-python-step-by-step-guide-485a83c79992>.

- [23] K. naik, Director, *Logistic Regression Multiclass Classification (OneVsRest)- Part 3/ Data Science*. [Film]. India: Krish naik, 2020.
- [24] R. Gandhi, "Support Vector Machine — Introduction to Machine Learning Algorithms," towardsdatascience.com/, 7 June 2018. [Online]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.
- [25] esarsouza, "Kernel Functions for Machine Learning Applications," crsouza.com, 17 Mach 2010. [Online]. Available: [http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/#kernel_methods%20\(SVM%20kernels\)](http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/#kernel_methods%20(SVM%20kernels)).
- [26] K. naik, Director, *Math's Intuition Behind Support Vector Machine Part 2*. [Film]. India: Krish Naik, 2020.
- [27] j. starmer, Director, *The Radial (RBF) Kernel*. [Film]. statquest, 2019.
- [28] r. Gandhi, "Naive Bayes Classifier," towards data science, 5 may 2018. [Online]. Available: <https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>.
- [29] G. Sanderson, Director, *Bayes Theorem*. [Film]. U.S: 3ble1brown, 2019.
- [30] S. Kim, "A Beginner's Guide to Convolutional Neural Networks (CNNs)," towards data science, 15 February 219. [Online]. Available: <https://towardsdatascience.com/a-beginners-guide-to-convolutional-neural-networks-cnns-14649dbddce8>.
- [31] K. S. a. A. Zisserman, "VGG-16," *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*, no. 2015, p. 14, 2014.
- [32] D. Godoy, "Understanding binary cross-entropy / log loss: a visual explanation," towards data science, 21 November 2018. [Online]. Available: <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>.
- [33] M. J. Matthew Yedlin, Director, *Binary Cross-Entropy*. [Film]. 2020.
- [34] V. Aliyev, "A definitive explanation to the Hinge Loss for Support Vector Machines.," towards data science, 24 November 2020. [Online]. Available: <https://towardsdatascience.com/a-definitive-explanation-to-hinge-loss-for-support-vector-machines-ab6d8d3178f1>.
- [35] *Hinge Loss - Machine Learning Algorithms: Supervised Learning Tip to Tail*. [Film]. 2020.
- [36] "Intuition of Adam Optimizer," geeks for geeks, 24 October 2020. [Online]. Available: <https://www.geeksforgeeks.org/intuition-of-adam-optimizer/#:~:text=Adam%20optimizer%20involves%20a%20combination,minima%20in%20a%20faster%20pace>.
- [37] R. Gandhi, "A Look at Gradient Descent and RMSprop Optimizers," towards data science, 20 June 2018. [Online]. Available: <https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizers-f77d483ef08b>.