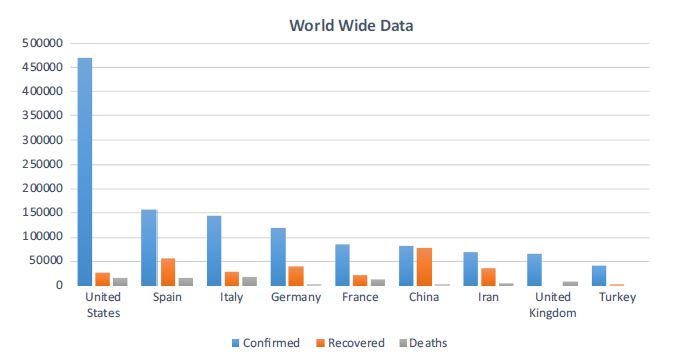
1. **INTRODUCTION**
   1. **OVERVIEW**

In December 2019, the novel coronavirus appeared in the Wuhan city of China and was reported to the World Health Organization (W.H.O) on 31st December 2019. The virus created a global threat and was named as COVID-19 by W.H.O on 11th February 2020. The COVID-19 is the family of viruses including SARS, ARDS. W.H.O declared this outbreak as a public health emergency and mentioned the following; the virus is being transmitted via the respiratory tract when a healthy person comes in contact with the infected person. The virus may transmit between persons through other roots which are currently unclear. The infected person shows symptoms within 2–14 days, depending on the incubation period of the middle east respiratory syndrome (MERS), and the severe acute respiratory syndrome (SARS). According to W.H.O the signs and symptoms of mild to moderate cases are dry cough, fatigue and fever while as in severe cases dyspnea (shortness of breath), Fever and tiredness may occur. The persons having other diseases like asthma, diabetes, and heart disease are more vulnerable to the virus and may become severely ill. The person is diagnoses based on symptoms and his travel history. Vital signs are being observed keenly of the client having symptoms. No specific treatment has been discovered as on 10th April 2020, and patients are being treated symptomatically. The drugs like hydroxychloriquine, antipyretic, anti-virals are used for the symptomatic treatment. Currently, no such vaccine is developed for preventing this deadly disease, and we may take some precautions to prevent this disease. By washing hands regularly with soap for 20 s and avoiding close contact with others by keeping the distance of about 1 m may reduce the chances of getting affected by this virus. While sneezing, Covering the mouth and nose with the help of disposable tissue and avoiding the contact with the nose, ear and mouth can help in its prevention. SARS is an airborne disease that appeared in 2003 in China and affected 26 countries by having 8 K cases in the same year and transferred from person to person. The signs and symptoms of SARS are fever, cold, diarrhoea, shivering, malaise, myalgia and dyspnea. The ARDS (acute respiratory distress syndrome) is characterized by rapid onset of inflammation in lungs which leads to respiratory failure and its signs and symptoms are bluish skin colour, fatigue and shortness of breath. ARDS is diagnosed by PaO2/FiO2 ratio of less than 300 mm Hg. Till 10th of April 2020, almost 1.6 million confirmed cases of coronavirus are detected around the globe. Almost 97 K persons have died and 364 K persons have recovered from this deadly virus. Figure 1 shows the worldwide data regarding coronavirus. Since no drug or vaccine is made for curing the COVID-19. Various paramedical companies have claimed of developing a vaccine for this virus. Less testing has also given rise to this disease as we lack the medical resources due to pandemic. Since thousands and thousands are being tested positive day by day around the globe, it is not possible to test all the persons who show symptoms.

Apart from clinical procedures, machine learning provides a lot of support in identifying the disease with the help of image and textual data. Machine learning can be used for the identification of novel coronavirus. It can also forecast the nature of the virus across the globe. However, machine learning requires a huge amount of data for classifying or predicting diseases. Supervised machine learning algorithms need annotated data for classifying the text or image into different categories. From the past decade, a huge amount of progress is being made in this area for resolving some critical projects. Recent pandemic has attracted many researchers around the globe to solve this problem.

Data provided by John Hopkins University in the form of X-ray images and various researchers build a model of machine learning that classifies X-ray image into COVID-19 or not. Since the latest data published by Johns Hopkins gives the metadata of these images. The data consists of clinical reports in the form of text in this paper, we are classifying that text into four different categories of diseases such that it can help in detecting coronavirus from earlier clinical symptoms. We used supervised machine learning techniques for classifying the text into four different categories COVID, SARS, ARDS and Both (COVID, ARDS). We are also using ensemble learning techniques for classification. Section 2 gives the literature survey regarding the proposed work. The framework for detecting coronavirus from clinical text data is being discussed in Sects. 3 and 4 gives the experimental results of the proposed framework and Sect. 5 concludes our work.

* 1. **PROBLEM SPECIFICATION**

Technology advancements have a rapid effect on every field of life, be it medical field or any other field. Artificial intelligence has shown the promising results in health care through its decision making by analyzing the data. COVID-19 has affected more than 100 countries in a matter of no time. People all over the world are vulnerable to its consequences in future. It is imperative to develop a control system that will detect the coronavirus. One of the solutions to control the current havoc can be the diagnosis of disease with the help of various AI tools

**1.3 METHODOLOGIES**

Machine learning and natural language processing use big data-based models for pattern recognition, explanation, and prediction. NLP has gained much interest in recent years, mostly in the field of text analytics, Classification is one of the major tasks in text mining and can be performed using different algorithms. Kumar et al. performed a SWOT analysis of various supervised and unsupervised text classification algorithms for mining the unstructured data. The various applications of text classification are sentiment analysis, fraud detection, and spam detection etc. Opinion mining is majorly being used for elections, advertisement, business etc. Verma et al. analyzed Sentiments of Indian government projects with the help of the lexicon-based dictionary. The machine learning has changed the perspective of diagnosis by giving great results to diseases like diabetes and epilepsy. Chakraborti et al. [9] detected epilepsy using machine learning approaches, electroencephalogram (EEG) signals are used for detecting normal and epileptic conditions using artificial neural networks (ANN). Sarwar et al. diagnosis diabetes using machine learning and ensemble learning techniques result indicated that ensemble technique assured accuracy of 98.60%. These purposes can be beneficial to diagnose and predict COVID-19. Firm and exact diagnosis of COVID-19 can save millions of lives and can produce a massive amount of data on which a machine learning (ML) models can be trained. ML may provide useful input in this regard, in particular in making diagnoses based on clinical text, radiography Images etc. According to Bullock et al., Machine learning and deep learning can replace humans by giving an accurate diagnosis. The perfect diagnosis can save radiologists’ time and can be cost-effective than standard tests for COVID-19. X-rays and computed tomography (CT) scans can be used for training the machine learning model. Several initiatives are underway in this regard. Wang and Wong developed COVID-Net, which is a deep convolutional neural network, which can diagnose COVID-19 from chest radiography images. Once the COVID-19 is detected in a person, the question is whether and how intensively that person will be affected. Not all COVID-19 positive patients will need rigorous attention. Being able to prognosis who will be affected more severely can help in directing assistance and planning medical resource allocation and utilization. Yan et al. used machine learning to develop a prognostic prediction algorithm to predict the mortality risk of a person that has been infected, using data from (only) 29 patients at Tongji Hospital in Wuhan, China. Jiang et al. proposed a machine learning model that can predict a person affected with COVID-19 and has the possibility to develop acute respiratory distress syndrome (ARDS). The proposed model resulted in 80% of accuracy. The samples of 53 patients were used for training their model and are restricted to two Chinese hospitals. ML can be used to diagnose COVID-19 which needs a lot of research effort but is not yet widely operational. Since less work is being done on diagnosis and predicting using text, we used machine learning and ensemble learning models to classify the clinical reports into four categories of viruses.

**1.4 CONTRIBUTIONS**

In this project we are using traditional and classical machine learning algorithms to predict COVID-19 disease. In traditional algorithms we are using Logistic Regression, Naïve Bayes, SVM and Decision Tree and in classical algorithms we are using Bagging, AdaBoost, Random Forest and Stochastic Gradient Boosting classifier. In all algorithms Logistic Regression giving better performance.

**LAYOUT THESIS**

**Data collection**

As W.H.O declared Coronavirus pandemic as Health Emergency. The researchers and hospitals give open access to the data regarding this pandemic. We have collected from an open-source data repository GitHub.1 In which about 212 patients’ data is stored which have shown symptoms of corona virus and other viruses. Data consists of about 24 attributes namely patient id, offset, sex, age, finding, survival, intubated, went\_icu, needed\_supplemental\_ O2, extubated, temperature, pO2\_saturation, leukocyte\_ count, neutrophil count, lymphocyte count, view, modality, date, location, folder, filename, DOI, URL. License. Clinical notes and other notes.

**Relevant dataset**

Since our work is regarding text mining so we extracted clinical notes and findings. Clinical notes consist of text while as the attribute finding consist label of the corresponding text. About 212 reports were used and their length was calculated. We consider only those reports that are written in the English language. Figure 3 gives the length distribution of clinical reports that are written in English. The clinical reports are labelled to their corresponding classes. In our dataset, we have four classes COVID, ARDS, SARS and Both (COVID, ARDS). Figure 4 shows the different classes in which clinical text is being categorized and corresponding report length.

**Preprocessing**

The text is unstructured so it needed to be refined such that machine learning can be done. Various steps are being followed in this phase; the text is being cleaned by removing unnecessary text. Punctuation and lemmatisation are being done such that the data is refined in a better way. Stopwords, symbols, Url’s, links are removed such that classification can be achieved with better accuracy. Figure 5 shows the main steps in preprocessing.

**1.5 EXISTING SYSTEM**

Machine learning and natural language processing use big data-based models for pattern recognition, explanation, and prediction. NLP has gained much interest in recent years, mostly in the field of text analytics, Classification is one of the major task in text mining and can be performed using different algorithms

Since the latest data published by Johns Hopkins gives the metadata of these images. The data consists of clinical reports in the form of text in this paper, we are classifying that text into four different categories of diseases such that it can help in detecting coronavirus from earlier clinical symptoms. We used supervised machine learning techniques for classifying the text into four different categories COVID, SARS, ARDS and Both (COVID, ARDS). We are also using ensemble learning techniques for classification.

**1.5.1 Disadvantages**

* It is imperative to develop a control system that will detect the coronavirus.
* People all over the world are vulnerableto its consequences in future.

**1.6 PROPOSED SYSTEM**

Proposed a machine learning model that can predict a person affected with COVID-19 and has the possibility to develop acute respiratory distress syndrome (ARDS). The proposed model resulted in 80% of accuracy. The samples of 53 patients were used for training their model and are restricted to two Chinese hospitals. ML can be used to diagnose COVID-19 which needs a lot of research effort but is not yet widely operational. Since less work is being done on diagnosis and predicting using text, we used machine learning and ensemble learning models to classify the clinical reports into four categories of viruses.

**1.6.1 Advantages**

1. We can predict a person affected with corona virus using machine learning model.
2. So that we can avoid the extreme problem in future
3. **LITERATURE SURVEY**

**2.1 CASE STUDY**

**Title**: **new coronavirus associated with human respiratory disease in China**

**Author**: Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh.

Emerging infectious diseases, such as severe acute respiratory syndrome (SARS) and Zika virus disease, present a major threat to public health1,2,3. Despite intense research efforts, how, when and where new diseases appear are still a source of considerable uncertainty. A severe respiratory disease was recently reported in Wuhan, Hubei province, China. As of 25 January 2020, at least 1,975 cases had been reported since the first patient was hospitalized on 12 December 2019. Epidemiological investigations have suggested that the outbreak was associated with a seafood market in Wuhan. Here we study a single patient who was a worker at the market and who was admitted to the Central Hospital of Wuhan on 26 December 2019 while experiencing a severe respiratory syndrome that included fever, dizziness and a cough. Metagenomic RNA sequencing4 of a sample of bronchoalveolar lavage fluid from the patient identified a new RNA virus strain from the family Coronaviridae, which is designated here ‘WH-Human 1’ coronavirus (and has also been referred to as ‘2019-nCoV’). Phylogenetic analysis of the complete viral genome (29,903 nucleotides) revealed that the virus was most closely related (89.1% nucleotide similarity) to a group of SARS-like coronaviruses (genus Betacoronavirus, subgenus Sarbecovirus) that had previously been found in bats in China5. This outbreak highlights the ongoing ability of viral spill-over from animals to cause severe disease in humans.

**Title: Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study.**

**Author**: A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram,

Background: In December, 2019, a pneumonia associated with the 2019 novel coronavirus (2019-nCoV) emerged in Wuhan, China. We aimed to further clarify the epidemiological and clinical characteristics of 2019-nCoV pneumonia. Methods: In this retrospective, single-centre study, we included all confirmed cases of 2019-nCoV in Wuhan Jinyintan Hospital from Jan 1 to Jan 20, 2020. Cases were confirmed by real-time RT-PCR and were analysed for epidemiological, demographic, clinical, and radiological features and laboratory data. Outcomes were followed up until Jan 25, 2020. Findings: Of the 99 patients with 2019-nCoV pneumonia, 49 (49%) had a history of exposure to the Huanan seafood market. The average age of the patients was 55·5 years (SD 13·1), including 67 men and 32 women. 2019-nCoV was detected in all patients by real-time RT-PCR. 50 (51%) patients had chronic diseases. Patients had clinical manifestations of fever (82 [83%] patients), cough (81 [82%] patients), shortness of breath (31 [31%] patients), muscle ache (11 [11%] patients), confusion (nine [9%] patients), headache (eight [8%] patients), sore throat (five [5%] patients), rhinorrhoea (four [4%] patients), chest pain (two [2%] patients), diarrhoea (two [2%] patients), and nausea and vomiting (one [1%] patient). According to imaging examination, 74 (75%) patients showed bilateral pneumonia, 14 (14%) patients showed multiple mottling and ground-glass opacity, and one (1%) patient had pneumothorax. 17 (17%) patients developed acute respiratory distress syndrome and, among them, 11 (11%) patients worsened in a short period of time and died of multiple organ failure. Interpretation: The 2019-nCoV infection was of clustering onset, is more likely to affect older males with comorbidities, and can result in severe and even fatal respiratory diseases such as acute respiratory distress syndrome. In general, characteristics of patients who died were in line with the MuLBSTA score, an early warning model for predicting mortality in viral pneumonia. Further investigation is needed to explore the applicability of the MuLBSTA score in predicting the risk of mortality in 2019-nCoV infection.

**Title: Text classification algorithms for mining unstructured data: a SWOT analysis**

**Author**: E. Bertino and N. Islam,

It has become increasingly crucial and imperative to facilitate knowledge extraction for decision support and deliver targeted information to analysts that span wide application domains. Interestingly, the buzzing term “big data” which is estimated to be 90% unstructured further makes it difficult to tap and analyze information with traditional tools. Text mining entails defining a process which transforms and substitutes this unstructured data into a structured one to discover knowledge. Use of classification algorithms to intelligently mine text has been studied extensively across literature. This study predominantly surveys the text classification algorithms employed in the process of mining unstructured data to report a conclusive analysis on the trend of their use in terms of their respective strengths, weaknesses, opportunities, and threats (SWOT). The scope of these algorithms is then explored apropos the application area of sentiment analysis, a typical text classification task. A mapping which determines the unexplored social media technologies and the extent of use of these algorithms within respective social media is proffered to give an insight to the amount of work that has been done in the domain of machine learning based sentiment analysis on social media.

**Title**: **Diagnosis of diabetes type-II using hybrid machine learning based ensemble model**

**Author**: C. Zhang and R. Green.

The work done in this paper exhibits an expert system-based ensemble model in diagnosing type-II diabetes. Diabetes Mellitus is a disease with high mortality rate that affects more than 60% population. The mindset of this task is to analyze various machine learning techniques for binary classification concerning with illness i.e., to diagnose whether a subject is suffering from disease or not. There are in total fifteen classifiers considered and out of them five major techniques namely: ANN, SVM, KNN, Naive Bayes and Ensemble are used. For achieving the desired goals, the tools that were employed namely matrix laboratory (MATLAB) and WEKA 3.6.13. In Ensemble method the predictive potentials of various individual classifiers are fused together. Using Ensemble method, it increases the performance by combining the classifying ability of individual classifiers and the chances of misclassifying a particular instance are reduced significantly, this provides a greater accuracy to the overall classification process. It is the enhancing technique that does the majority voting and gives us the percolated results. The medical database analysed in this study includes a rich database of about 400 people from across a wide geographical region and ten physiological attributes. Furthermore, this diagnostic tool is examined by verifying denary cross attestation; on top of that the outcome has been confronted along the truly existing real interpretation about the cases. A GUI based diagnostic tool founded upon ensemble classifier is developed in such a manner it would be able to predict whether a patient is enduring against the disease or not when it is fed with all the 10 attributes from user through a user-friendly GUI (Graphical User Interface). The development of this diagnostic tool is done using MATLAB 2013a. Out of 10 parameters that the user needs to enter as input in GUI based diagnostic tool five are numeric and the rest are nominal values. The diagnostic tool in execution is demonstrated below in Fig. 3. The main objective of this manuscript is to propose an intelligent framework that will act as a useful aid for doctors for correct and timely biopsy can be done at early stage. The result indicated that ensemble technique assured an accuracy of 98.60% that clubs the predictive performance of multiple AI based algorithms and are superior in comparison with all other individual counterparts. The algorithms with better exactness than others are followed by Artificial neural network (ANN), Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN).

**2.2 FEASIBILITY STUDY**

**Feasibility Study:**

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility
* Economical Feasibility

**2.2.1 TECHNICAL FEASIBILITY**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

**2.2.2 OPERATIONAL FEASIBILITY**

**User-friendly**

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. Theses forms and reports are generated as user-friendly to the Client.

**Reliability**

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

**Security**

The web server and database server should be protected from hacking, virus etc

**Portability**

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

**Availability**

This software will be available always.

**Maintainability**

The system called the ewheelz uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-SQL, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

**2.2.3 ECONOMIC FEASILITY**

The computerized system takes care of the present existing system’s data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web-based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

Open-source software like TOMCAT, JAVA, MySQL and WINDOWS is used to minimize the cost for the Customer.

1. **SYSTEM REQUIREMENTS**

**3.1 SYSTEM REQUIREMENT ANALYSIS**

**3.1.1 SOFTWARE REQUIREMENTS**

Programming language : Python

Operating System : Windows 10

**3.1.2 HARDWARE REQUIREMENTS**

Preprocessor : AMD Ryzen 3 (2.6 GHz CPU Speed, 2 Cores)

RAM : 8GB DDR4

Hard Disk : 1 TB

1. **SYSTEM DESIGN**

**4.1 UML DIAGRAMS**

4.1.1 Use case Diagram



4.1.1 Use case diagram

4.1.2 Class Diagram



4.1.2 Class diagram

4.1.3 Sequence Diagram



4.1.3 Sequence diagram

4.1.4 Collaboration Diagram



4.1.4 Collaboration Diagram

**4.2 DESIGNS**

4.2.1 Architecture Design



Fig 4.2.1 Architecture Diagram

4.2.2 Dataflow Diagram

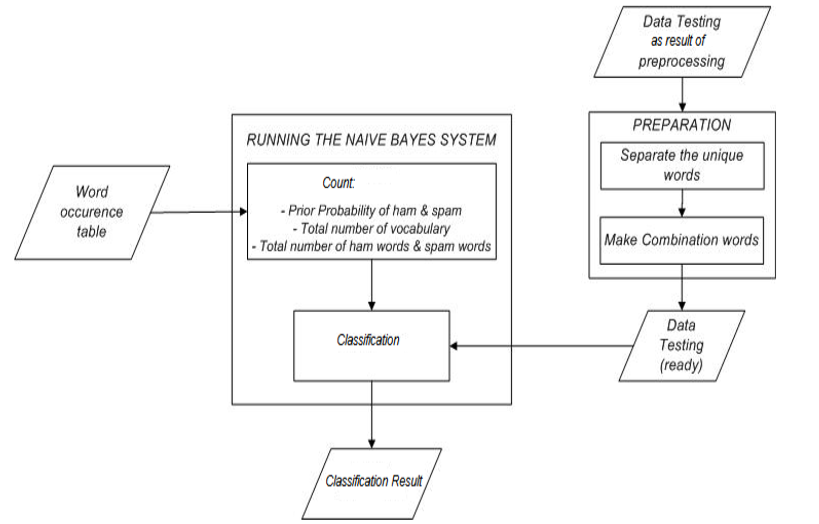


Fig 4.2.2 Dataflow Diagram

**4.3 FLOW CHART**

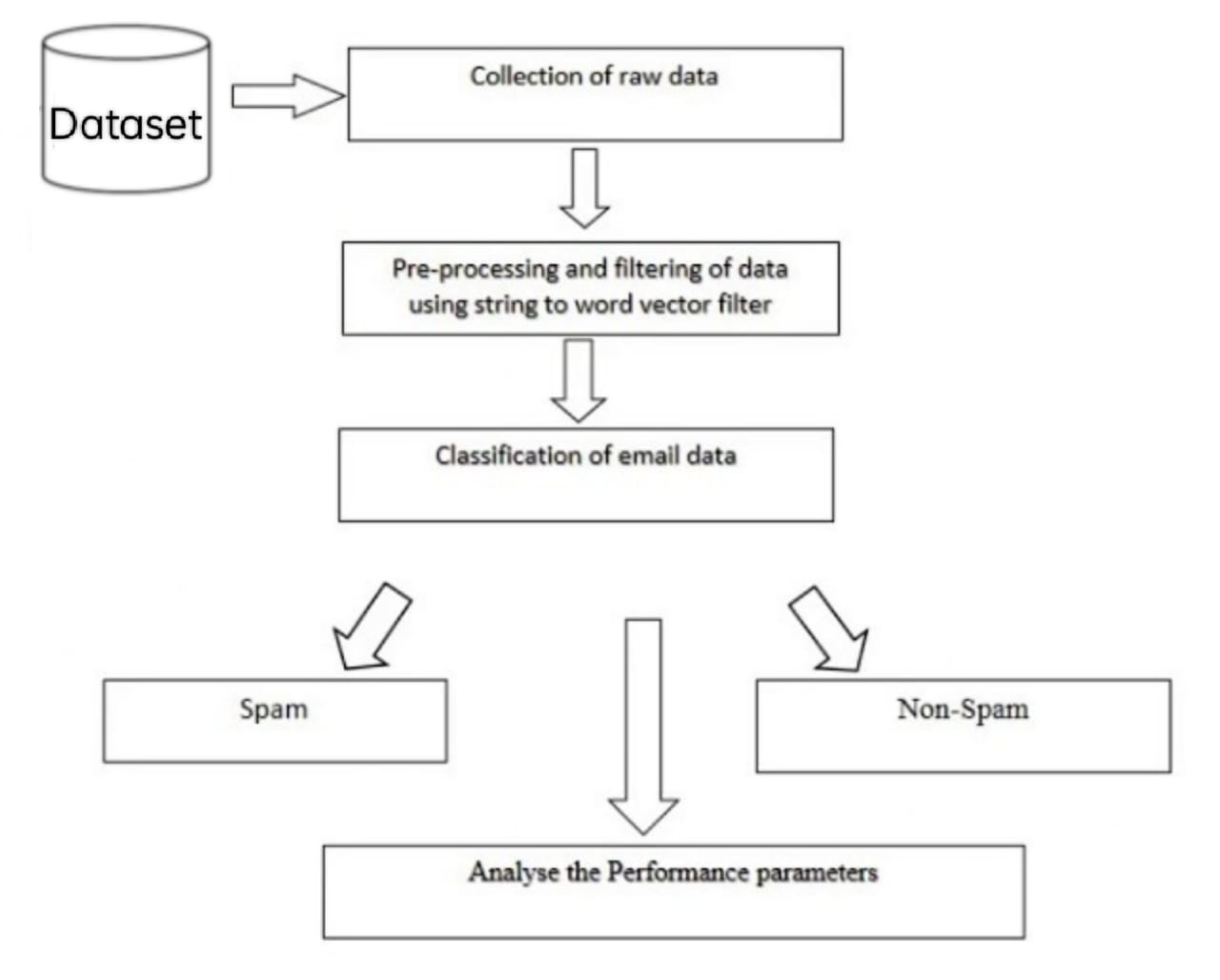


Fig 4.3 Flow Chart

1. **TECHNOLOGIES**

**5.1 PYTHON**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An [interpreted language](https://en.wikipedia.org/wiki/Interpreted_language), Python has a design philosophy that emphasizes code [readability](https://en.wikipedia.org/wiki/Readability) (notably using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block) rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) than might be used in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B)or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is [open source](https://en.wikipedia.org/wiki/Open_source) software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation). Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming), and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

**5.2 DJANGO**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself). Python is used throughout, even for settings files and data models.

Fig 5.2 Django

Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)) and configured via admin models

1. **IMPLEMENTATION**

**6.1 INTRODUCTION**

This dataset contains more than 30 columns but we are extracting two column values such as ‘clinical\_notes’ and ‘finding’. clinical\_notes column contains medical text data and this text data is preprocess using NLTK library to remove stop words, special symbols and then apply lemmatizer to remove ‘ing, tion etc.’ from text. After preprocess text we will apply TF-IDF to extract top 40 features from dataset. Below is the dataset screen shots and this dataset saved inside ‘dataset’ folder.

‘Upload Covid-19 Dataset’ button and then upload dataset

* selecting and uploading ‘dataset.csv’ file and then click on ‘Open’ button to load dataset
* we extract all text data from dataset and now in above screen text in first sentence we have ‘on’ stop words and many number of numerical values and to remove those stop words and to clean data
* preprocess all stop words removed out and in above ‘on’ stop word removed out
* above graph showing count/finding of each label and now close above graph and then click on ‘Feature Engineering’ button to apply TF-IDF on above text data and to get below features
* screen all text data converted to above TF-IDF features and now click on ‘Run Logistic Regression, Naive Bayes, SVM & Decision Tree’ to run all traditional algorithms on features data and to calculate accuracy

Displaying accuracy, precision, recall and FScore for each algorithm and now click on ‘Run Bagging, Adaboost, Random Forest & Gradient Boosting’ button to calculate accuracy of classical algorithms

* showing classical algorithms accuracy and other metrics values and now click on ‘Comparative Analysis Graph’ button to get below graph
* we can see accuracy, precision, recall and fscore for each algorithm in group bar chart and in above graph x-axis represents algorithm name y-axis represents values

**6.2 DOMAIN DESCRIPTION**

**What is Machine Learning? A Definition.**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, **an approach based on semantic analysis mimics the human ability to understand the meaning of a text.**

**6.2.1 Some Machine Learning Methods**

Machine learning algorithms are often categorized as supervised or unsupervised.

* **Supervised machine learning algorithms**can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
* In contrast, **unsupervised machine learning algorithms**are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
* **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from

it. Otherwise, acquiring unlabeled data generally doesn’t require additional resources.

* **Reinforcement machine learning algorithms**is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

**6.3 SAMPLE CODE**

import re

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

import matplotlib.pyplot as plt

from matplotlib import pyplot as plt

from nltk import ngrams

import numpy as np

import textblob

import re, nltk

from nltk.stem import WordNetLemmatizer

import nltk

import collections

from nltk.corpus import stopwords

#directly downloads packages for required things

nltk.download('punkt')

nltk.download('popular')

from nltk.corpus import stopwords

import pandas as pd

import string

#importing the data

from collections import Counter

D={}

wordnet\_lemmatizer = WordNetLemmatizer()

stop\_words = set(stopwords.words('english'))

def read\_data():

raw\_data = pd.read\_csv(r'covid.csv', engine='python')

#raw\_data=raw\_data['text']

return raw\_data

def normalizer(tweet):

aplhabets = re.sub("[^a-zA-Z]", " ",tweet)

tokens = nltk.word\_tokenize(aplhabets)[2:]

lower\_case = [l.lower() for l in tokens]

processed\_words = list (filter(lambda l: l not in stop\_words, lower\_case))

#print(processed\_words)

lemmas = [wordnet\_lemmatizer.lemmatize(t) for t in processed\_words]

return lemmas

def sentiment(tweet):

return textblob.TextBlob(tweet).sentiment.polarity

def feauture\_labelling(score):

if (score >= 1):

return 5

elif (score <= -1):

return 1

elif ((score > -0.5) and (score < 0.5)):

return 3

elif ((score >= 0.5) and (score < 1)):

return 4

elif ((score <= -0.5) and (score > -1)):

return 2

def preprocess(test\_data):

# pass the test dataset here

print("given size of the data", len(test\_data))

# dropping the null values

print(test\_data.head())

test\_data = test\_data.drop\_duplicates()

print(" size of data after removing repitations", len(test\_data))

# print(test\_data['TWEETS'].value\_counts())

# DATA PRECPROCESSING STARTS FROM HERRE

# create a coloum in the dataset to get only normalized words

test\_data["normalized\_tweet"] = np.nan

for i in test\_data:

print(i)

test\_data['normalized\_tweet'] = test\_data["text"].apply(lambda x: re.split('https:\/\/.\*', str(x))[0])

test\_data["normalized\_tweet"] = test\_data["normalized\_tweet"].apply(lambda x: normalizer(x))

print(test\_data["normalized\_tweet"])

# now we are converting to ngrams to get the meanining

test\_data["processed\_tweet"] = np.nan

test\_data["sentiment"] = np.nan

test\_data["processed\_tweet"] = test\_data["normalized\_tweet"].apply(lambda x: " ".join(list(x)))

print(test\_data["processed\_tweet"].head())

test\_data["sentiment"] = test\_data["processed\_tweet"].apply(lambda x: sentiment(x))

print(test\_data["sentiment"].value\_counts())

test\_data["score"] = test\_data["sentiment"].apply(lambda x: feauture\_labelling(x))

test\_data.to\_csv('..\preprocessed\_reviews2.csv')

return test\_data

def naivebayes(test\_data):

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

test\_data= test\_data[['sentiment', 'score']]

X = test\_data.drop(columns=['score'])

print("\n \*\*\*\*\*\*\*\*dimension of input trained data\*\*\*\*\*\*\*\n ", X.shape)

Y = test\_data['score'].astype("int")

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state=1)

from sklearn.naive\_bayes import GaussianNB

# Create a nb Classifier

clf = GaussianNB()

# print(x\_train.value\_counts())

# Train the model using the training sets

clf.fit(x\_train, y\_train)

# Predict the response for test dataset

y\_pred = clf.predict(x\_test)

# print(y\_pred)

# model evaluation

from sklearn import metrics

# Model Accuracy: how often is the classifier correct?

dft = pd.DataFrame()

dft = x\_test

dft['predictions'] = list(y\_pred)

print("Accuracy for NAIVE BAYES :", metrics.accuracy\_score(y\_test, y\_pred))

D["Naive\_bayes"]=metrics.accuracy\_score(y\_test, y\_pred)

print("plotting the distrubution in test split")

dft['predictions'].value\_counts().sort\_index().plot(kind='bar', figsize=(20, 15),

title='NAIVE BAYES prediction distrubution in test data',

color=['cyan','teal'])

plt.xlabel("PREDICTED CLASSES")

plt.ylabel('COUNT')

plt.show()

y\_pred = clf.predict(x\_test)

def supportvm(test\_data):

from sklearn.svm import SVC

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

test\_data = test\_data[['sentiment', 'score']]

X = test\_data.drop(columns=['score'])

print("\n \*\*\*\*\*\*\*\*dimension of input trained data\*\*\*\*\*\*\*\n ", X.shape)

Y = test\_data['score'].astype("int")

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state=1)

from sklearn.naive\_bayes import GaussianNB

# Create a nb Classifier

clf = SVC(kernel='poly')

# print(x\_train.value\_counts())

# Train the model using the training sets

clf.fit(x\_train, y\_train)

# Predict the response for test dataset

y\_pred = clf.predict(x\_test)

# print(y\_pred)

# model evaluation

from sklearn import metrics

# Model Accuracy: how often is the classifier correct?

dft = pd.DataFrame()

dft = x\_test

dft['predictions'] = list(y\_pred)

print("Accuracy for SVC :", metrics.accuracy\_score(y\_test, y\_pred))

D["Support\_vector"]= metrics.accuracy\_score(y\_test, y\_pred)

print("plotting the distrubution in test split")

dft['predictions'].value\_counts().sort\_index().plot(kind='bar', figsize=(20, 15),

title='SVC prediction distrubution in test data',

color=['red','blue'])

plt.xlabel("PREDICTED CLASSES")

plt.ylabel('COUNT')

plt.show()

if \_\_name\_\_=="\_\_main\_\_":

raw\_data=read\_data()

p\_df=preprocess(raw\_data)

supportvm(p\_df)

naivebayes(p\_df)

plt.title("naive bayes accuracy: {0} vs support vector machines accuracy{1}".format(round(D["Naive\_bayes"],4),round(D["Support\_vector"],4)))

plt.bar(\*zip(\*D.items()))

plt.show()

1. **TESTING ANALYSIS**

**7.1 SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 7.2 TYPES OF TESTS

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Id** | **Test Name** | **Input** | **Output** | **Expected Result** | **Status** |
| 1 | Covid-19 Collect dataset | file | Data stored | Data stored | PASS |
| 2 | Master browsing data | data | Data is not Present in directory | Data stored | FAIL |
| 3 | Analysis Algorithm | File Name with Uploaded data | File Attacked or Safe | Attacked or Safe | PASS |
| 4 | Comparative Analysis Graph’ | No Inputs | No Spam to Compare File | Comparative Analysis Graph’ | PASS |
|  |  |  |  |  |  |

**7.3 TESTING CASES**

1. **SCREEN SHOTS**

**8.1 Data**

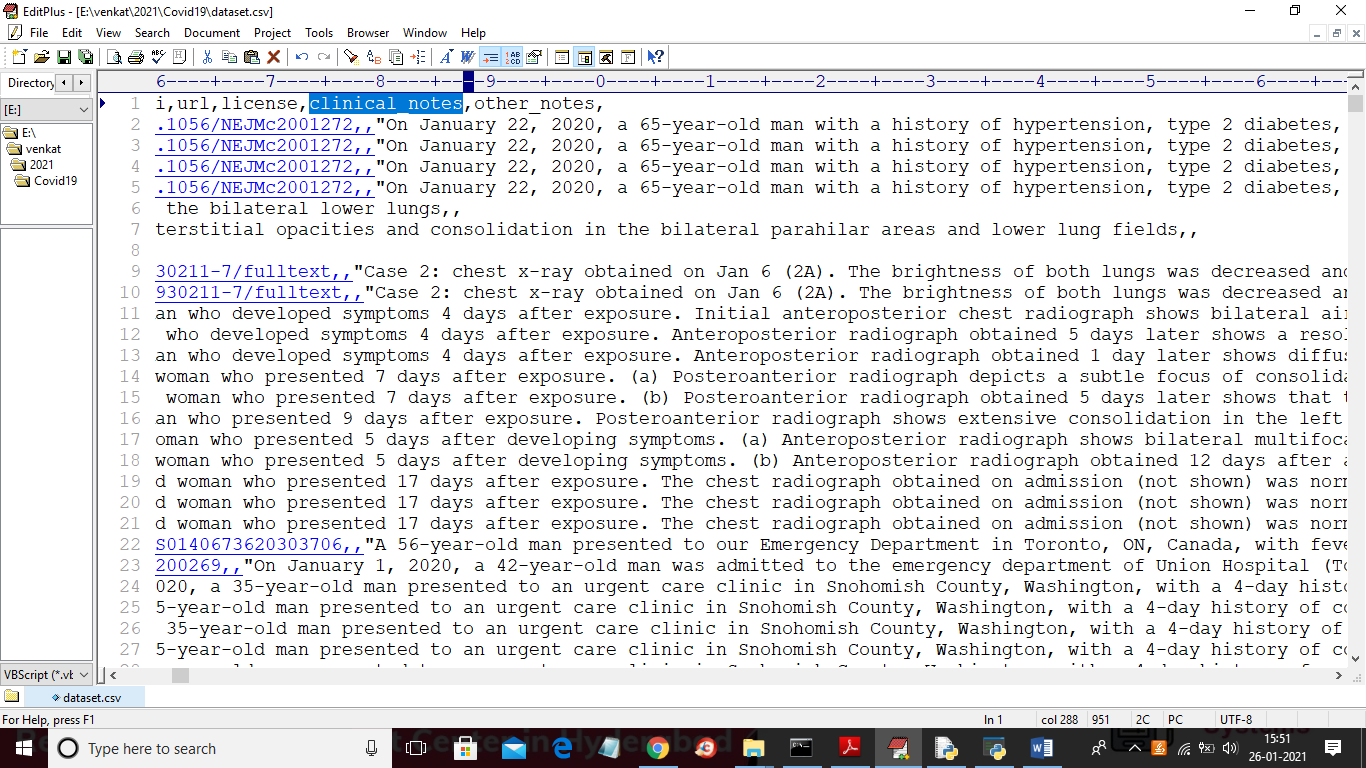


Fig 8.1 Data

**8.2 Interface**

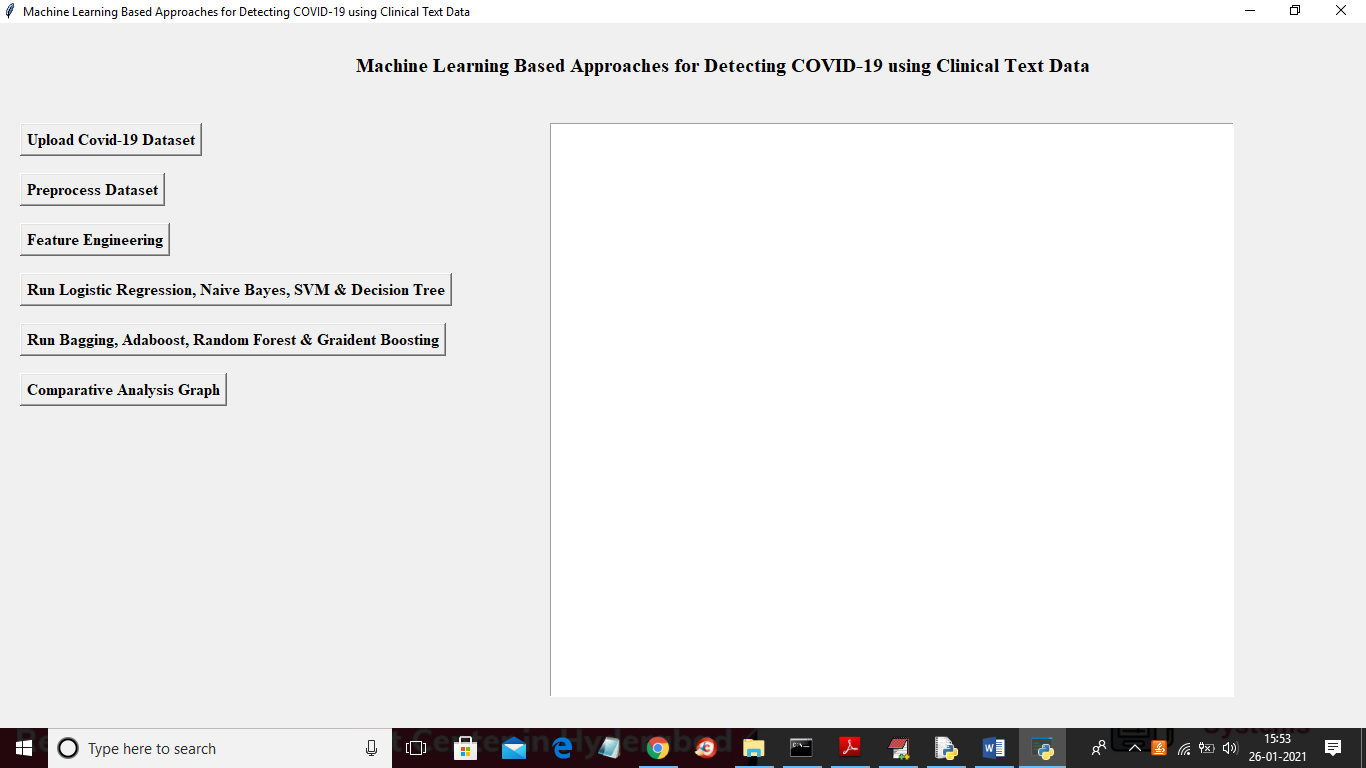
****

Fig 8.2 Interface

**8.3 Uploading dataset**

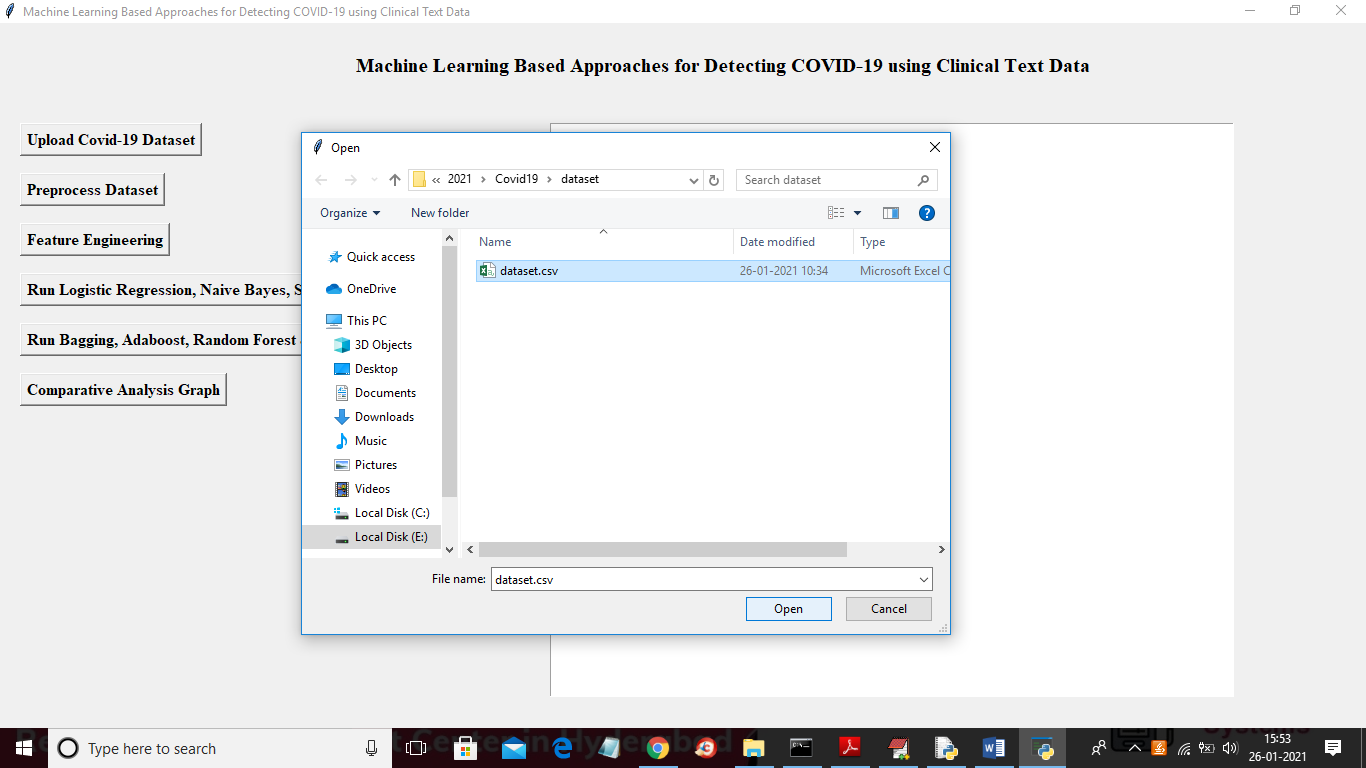
****

Fig 8.3 uploading dataset

**8.4 Reading dataset**

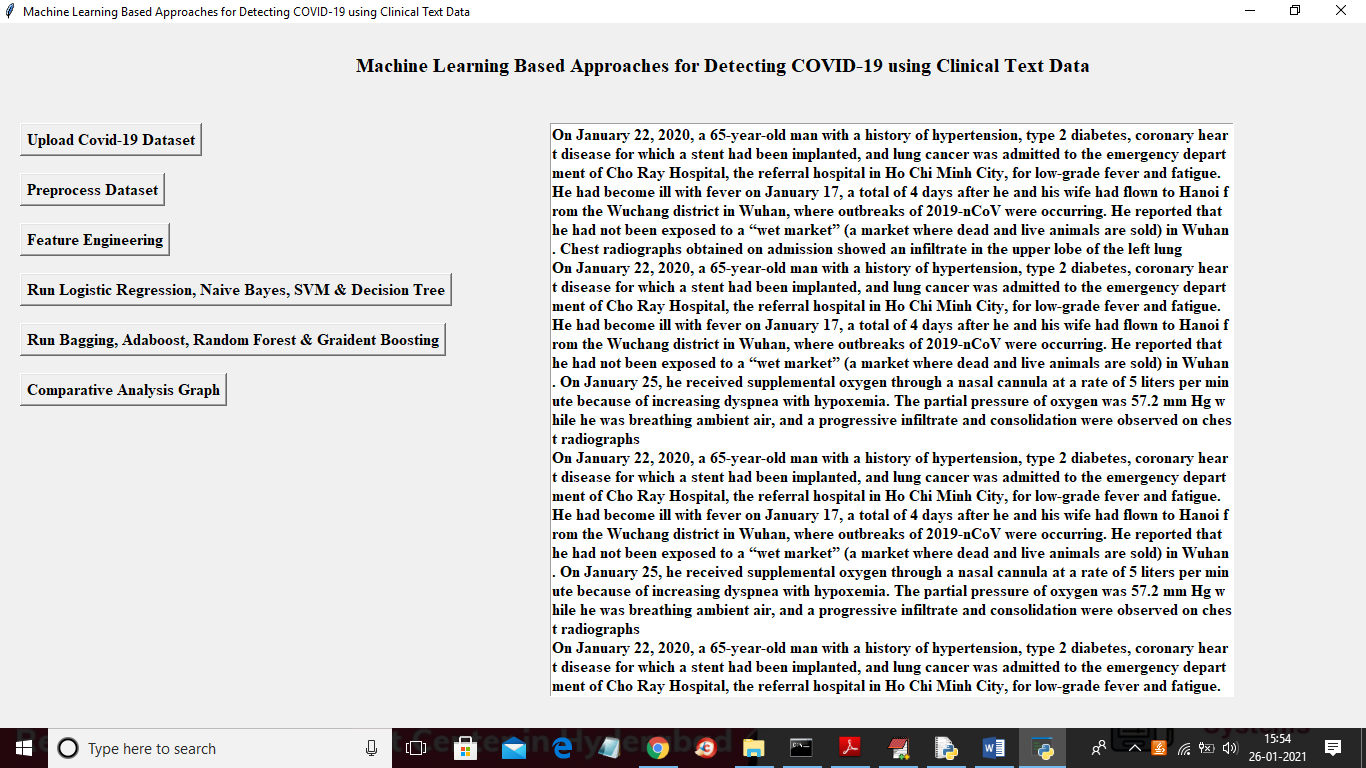
****

Fig 8.4 Reading dataset

**8.5 Preprocessing Dataset**

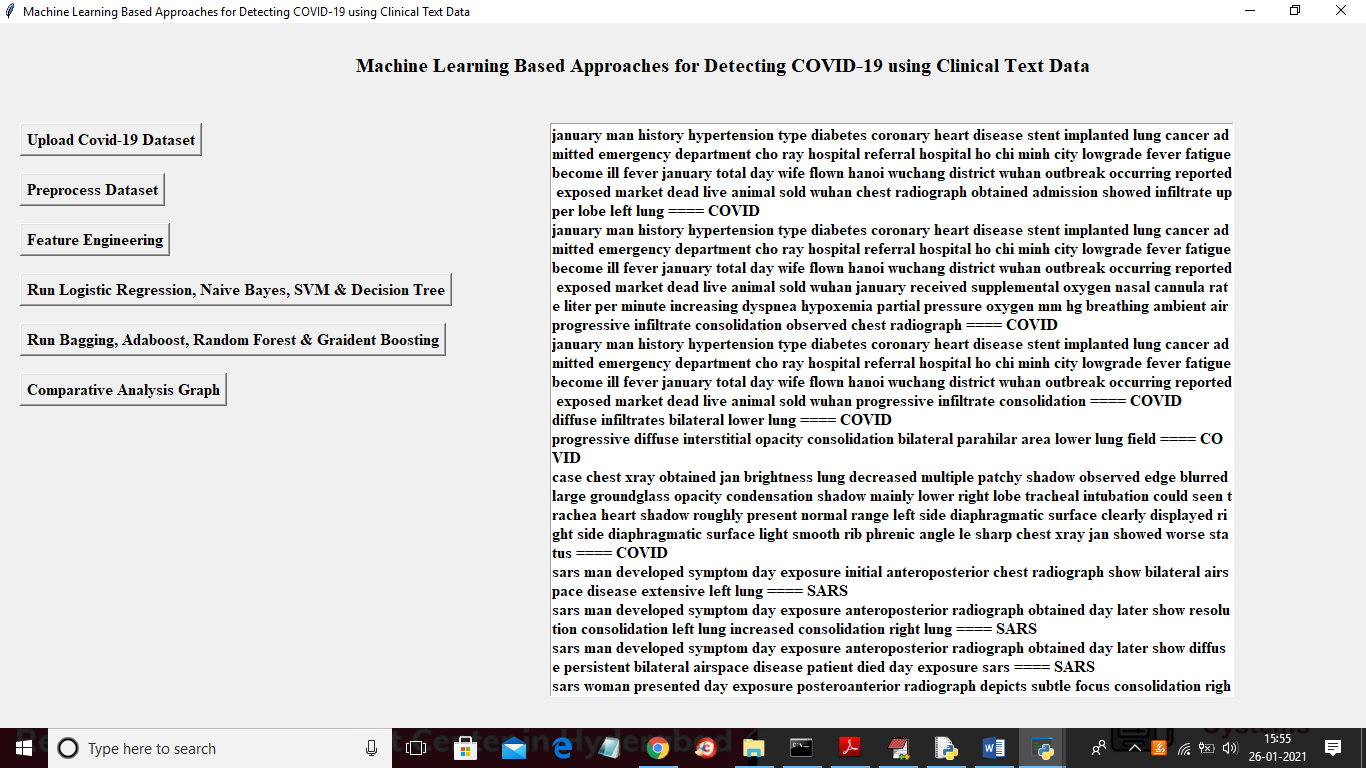
****

Fig 8.5 Preprocessing Dataset

**8.6 View Graph**

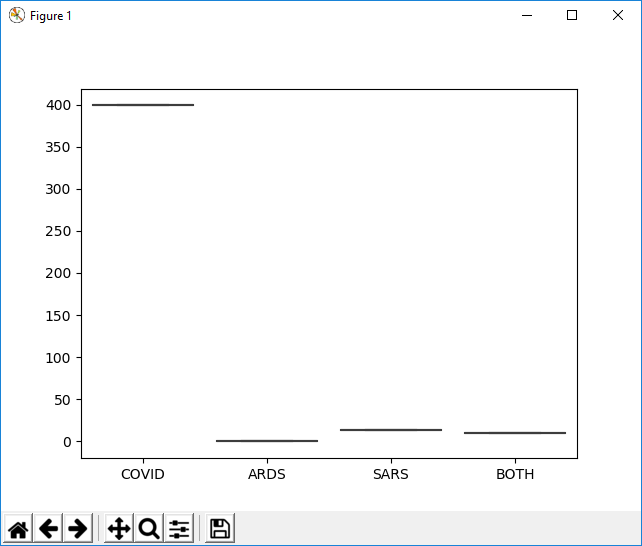
****

Fig 8.6 View Graph

**8.7 Feature Engineering**

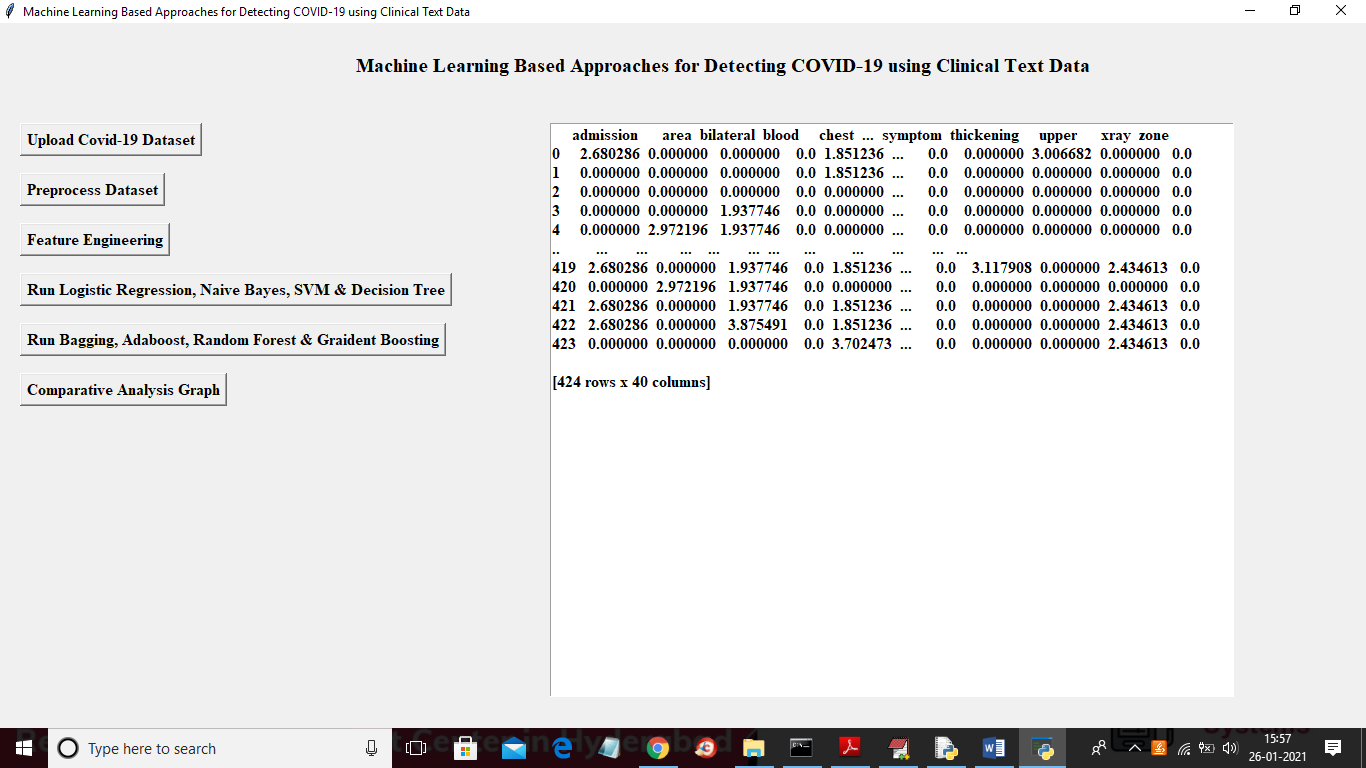
****

Fig 8.7 Feature Engineering

**8.8 Run logic Regression, Naive bayes, SVM and Decision Tree**

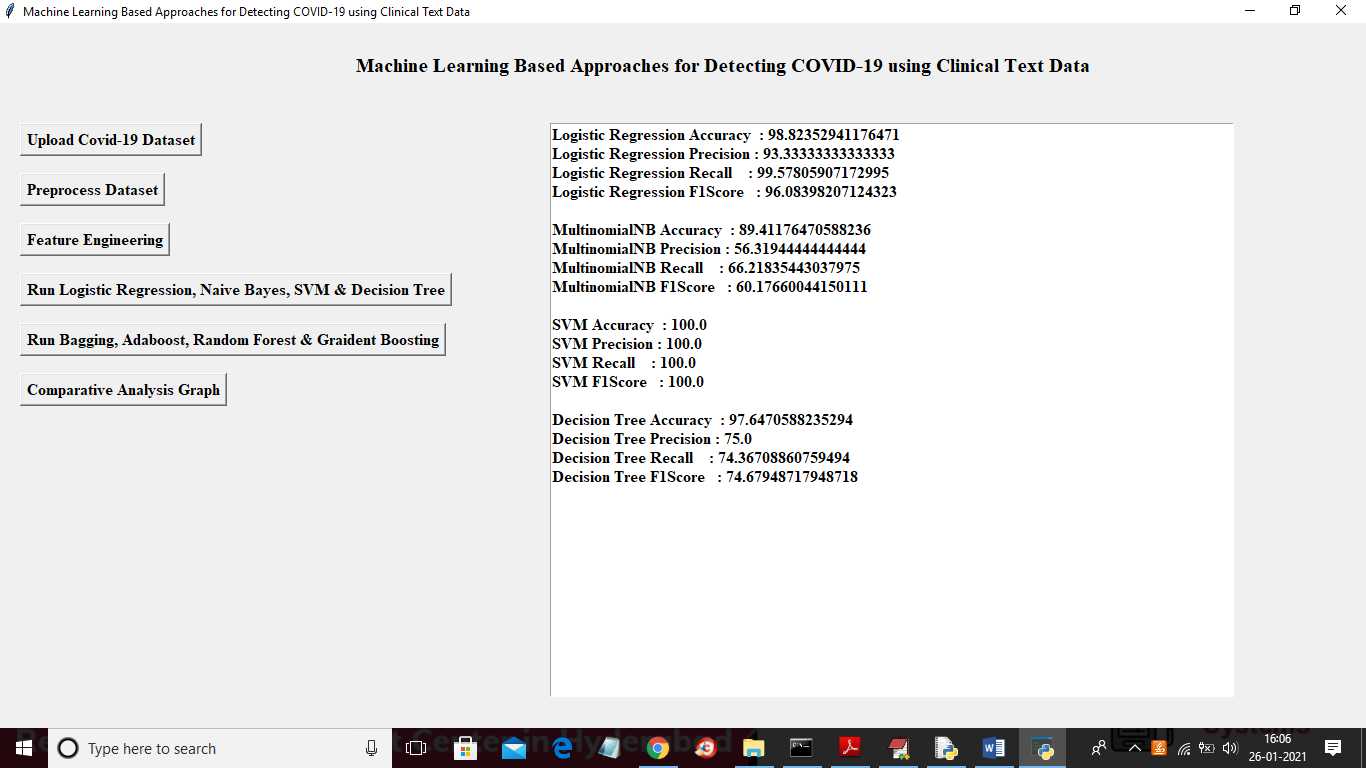
****

Fig 8.8 Run logic regression, naive bayes, SVM & Decision Tree

**8.9 Bagging Classifier, AdaBoost, Random Forest and Gradient**

**Boosting**

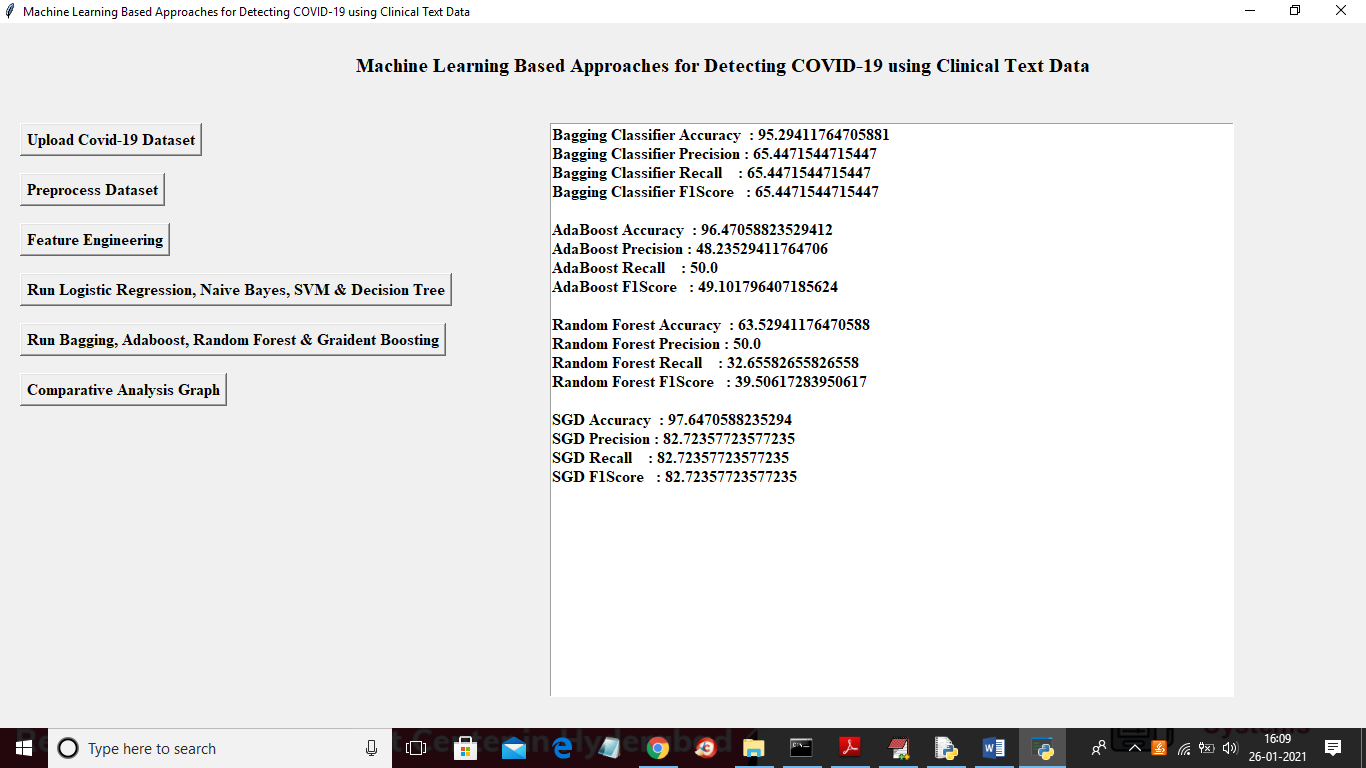
****

Fig 8.9 Bagging Classifier, AdaBoost, Random Forest & Gradient Boosting

**8.10 Comparative Graph**

****

Fig 8.10 Comparative Graph

1. **FUTURE SCOPE AND CONCLUSION**

COVID-19 has shocked the world due to its non-availability of vaccine or drug. Various researchers are working for conquering this deadly virus. We used 212 clinical reports which are labelled in four classes namely COVID, SARS, ARDS, and both (COVID, ARDS). Various features like TF/IDF, bag of words is being extracted from these clinical reports. The machine learning algorithms are used for classifying clinical reports into four different classes. After performing classification, it was revealed that logistic regression and multinomial Naïve Bayesian classifier gives excellent results by having 94% precision, 96% recall, 95% f1 score and accuracy 96.2%. Various other machine learning algorithms that showed better results were random forest, stochastic gradient boosting, decision trees and boosting. The efficiency of models can be improved by increasing the amount of data. Also, the disease can be classified on the gender-based such that we can get information about whether male are affected more or females. More feature engineering is needed for better results and deep learning approach can be used in future.

1. **REFERENCES**

1. Wu F, Zhao S, Yu B, Chen YM, Wang W, Song ZG, Hu Y, Tao ZW, Tian JH, Pei YY, Yuan ML, Zhang YL, Dai FH, Liu Y, Wang QM, Zheng JJ, Xu L, Holmes EC, Zhang YZ (2020) A new coronavirus associated with human respiratory disease in china. Nature 44(59):265–269

2. Medscape Medical News, The WHO declares public health emergency for novel coronavirus (2020) https://www.medscape. com/viewarticle/924596

3. Chen N, Zhou M, Dong X, Qu J, Gong F, Han Y, Qiu Y, Wang J, Liu Y, Wei Y, Xia J, Yu T, Zhang X, Zhang L (2020) Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study. Lancet 395(10223):507–513

4. World health organization: https://www.who.int/new-room/g-adetail/ q-a-corronaviruses#:/text=symptoms. Accessed 10 Apr 2020

5. Wikipedia coronavirus Pandemic data: https://en.m.wikipedia. org/wiki/Template:2019%E2%80%9320\_coronavirus\_pan demic\_data. Accessed 10 Apr 2020

6. Khanday, A.M.U.D., Amin, A., Manzoor, I., & Bashir, R., ‘‘Face Recognition Techniques: A Critical Review’’ 2018

7. Kumar A, Dabas V, Hooda P (2018) Text classification algorithms for mining unstructured data: a SWOT analysis. Int J Inf Technol. <https://doi.org/10.1007/s41870-017-0072-1>

8. Verma P, Khanday AMUD, Rabani ST, Mir MH, Jamwal S (2019) Twitter Sentiment Analysis on Indian Government Project using R. Int J Recent Tech Eng. [https://doi.org/10.35940/ijrte. C6612.098319](https://doi.org/10.35940/ijrte.%20C6612.098319)

9. Chakraborti S, Choudhary A, Singh A et al (2018) A machine learning based method to detect epilepsy. Int J Inf Technol 10:257–263. <https://doi.org/10.1007/s41870-018-0088-1>

10. Sarwar A, Ali M, Manhas J et al (2018) Diagnosis of diabetes type-II using hybrid machine learning based ensemble model. Int J Inf Technol. <https://doi.org/10.1007/s41870-018-0270-5>

11. Bullock J, Luccioni A, Pham KH, Lam CSN, Luengo-Oroz M (2020) Mapping the landscape of artificial intelligence applications against COVID-19. <https://arxiv.org/abs/2003.11336v1>

12. Wang L, Wong A (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 Cases from chest radiography images. <https://arxiv.org/abs/2003.09871>

13. Yan L, Zhang H-T, Xiao Y, Wang M, Sun C, Liang J, Li S, Zhang M, Guo Y, Xiao Y, Tang X, Cao H, Tan X, Huang N, Amd A, Luo BJ, Cao Z, Xu H, Yuan Y (2020) Prediction of criticality in patients with severe covid-19 Infection using three clinical features: a machine learning-based prognostic model with clinical data in Wuhan. medRxiv. [https://doi.org/10.1101/2020.02.27. 20028027](https://doi.org/10.1101/2020.02.27.%2020028027)

14. Jiang X, Coffee M, Bari A, Wang J, Jiang X, Huang J, Shi J, Dai J, Cai J, Zhang T, Wu Z, He G, Huang Y (2020) Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. Compu Mater Contin 63(1):537–551

15. Description of Logistic Regression Algorithm. https://machine learningmastery.com/logistic-regression-for-machine-learning/. Accessed 15 May 2019

16. Description of Multinomial Naı¨ve Bayes Algorithm https://www. 3pillarglobal.com/insights/document-classification-using-multi nomial-naive-bayes-classifier. Accessed 15 May 2019

17. Khanday AMUD, Khan QR, Rabani ST. SVMBPI: support vector machine based propaganda identification. SN Appl. Sci. (accepted)

18. Description of Decision Tree Algorithm: https://dataspirant.com/ 2017/01/30/how\_decision\_tree\_algorithm\_works/. Accessed 10 July 2019

19. Description of Boosting Algorithm: https://towardsdatascience. com/boosting. Accessed 10 July 2019

20. Description of Adaboost Algorithm: https://towardsdatascience. com/boosting-algorithm-adaboost-b673719ee60c. Accessed 10 July 2019