Predicting COVID-19 Disease Outcome and Post-Recovery Conditions using Machine Learning

by

Abul Kasem Sajid
21141066
Fahim Kabir
17101186
Hasibur Rahman
17201024
Indronil Kundu
17201013
Sheersho Zaman
21141079

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sajid

Abul Kasem Sajid

21141066

Fahim Kabir

17101186

Hasibur Rahman

17201024

Indronil Kundu

17201013

Zaman

Sheersho Zaman

21141079

Approval

The thesis/project titled "Predicting COVID-19 Disease Outcome and Post-Recovery Conditions using Machine Learning" submitted by

1. Abul Kasem Sajid (21141066)
2. Fahim Kabir (17101186)
3. Hasibur Rahman (17201024)
4. Indronil Kundu (17201013)
5. Sheersho Zaman (21141079)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 02 2021.

Examining Committee:

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department Of Computer Science And Engineering
Brac University

Dr. Md. Golam Rabiul Alam
Associate Professor
Department Of Computer Science And Engineering
Brac University

SHKazi

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

With COVID-19 still running rampant across the world, accurate diagnosis of patients and proper management of medical resources is paramount in order to deliver proper care to those that need it most. In order to do this, prediction models with the help of various machine learning algorithms are being developed across the world. Each may deal with certain variables that help predict the disease outcome, such as comorbidities, symptoms, age, sex, etc. Some models have also been made to help predict the chances of a COVID-19 patient in developing lasting medical conditions post recovery. The goal of this research then, is to create a model that takes all the aforementioned dimensions into account and create a prediction model with the three timelines in mind. It is a model that will predict if a person has contacted COVID-19 based on the preliminary symptoms they show (Timeline 1), predict the chances of a COVID-19 patient developing more serious symptoms based on their medical history (Timeline 2) and also predict the chances of a patient developing post-recovery conditions arising after recovering from COVID-19 (Timeline 3). To accomplish this, we use three machine learning algorithms – Random Forest, Naïve Bayes and K-nearest Neighbors. For implementation and testing of the model, data on COVID-19 patients is split into train and test sets and fit over the aforementioned algorithms. Their performance are then evaluated. Specific features of the dataset also analyzed at a deeper level in order to gain a better understanding of how the virus behaves in certain conditions. Having such a model in place will not only help us direct medical resources to patients that need the most attention, but will also provide a clearer understanding of the nature of the virus and how it affects a specific patient.

Keywords: Symptoms; Machine Learning; COVID-19; Prediction; ICU; Emergency; Random Forrest; Naïve Bayes; KNN

Acknowledgement

First, all praise to the Almighty, by whose grace we were able to complete our thesis during these trying times without any major interruption.

Next, we would like to thank our advisor, Dr. Muhammad Iqbal Hossain, for his continued support and guidance. He was always readily available to help us at a moment's notice and provided valuable insight that helped guide this thesis in the right direction. Finally, we must extend our gratitude to our respective families. The constant support and motivation they provided to each of us throughout this journey helped us immeasurably to reach this point. We would not have gotten this far without them.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Coronavirus or COVID-19 is caused by a newly discovered severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), with initial infections appearing towards the end of 2019 in Wuhan City, China. As of March 2020, the outbreak of COVID-19 has been declared as a global pandemic and has proven to be a struggle for health care systems around the world, as they fail to cope with the exponential increase in the number of affected. With asymptomatic human to human spreading and the overall degree to which the virus itself spreads when in the proximity of an infected person, containment of the virus has been challenging around the globe.

By the end of September 2020, there have been over 32.6 million confirmed cases of COVID-19 infection globally, with almost a million deaths over the world associated with the infectious disease. As of December 2020, there has been a total of 1.88 million deaths worldwide, with 87.2 million cases. Though the number of deaths rate itself was initially comparable to that of the flu, its contagious nature coupled with how difficult it is to detect and treat it make it incredibly dangerous. Even with successful testing, the virus affects people in varying degrees, with some people experiencing mild symptoms of a sore throat and fatigue for example, to more extreme cases that result in respiratory assistance via ventilators or worse.

One of the main challenges of dealing with COVID-19 is how diverse it is in terms of its severity in those affected. Being categorized in a spectrum [11], there are so many variations to the strand of the virus that some cases begin and remain asymptomatic while others escalate into severe conditions requiring serious medical attention. Generally, COVID-19 symptoms include high fever, feeling weak, headache, dry cough, sputum production, haemoptysis, diarrhea and also loss of smell and taste during the primary stages of infection. It is these symptoms which can quickly develop into acute respiratory syndrome (ARS), resulting in the patient requiring immediate critical medical attention. If they do not get treated in time or sometimes even if they do, the condition of the patient worsens to the point of death.

1.1 Motivation

With the COVID-19 pandemic being as contagious as it is, one of the most pressing concerns during its initial outbreak (alongside containing it) was the lack of testing ability that presented itself. The first step to containing and subsequently treating the virus was finding out the number of people who were affected and giving them the proper medical attention they needed, along with isolating them to stop the spread. Since the virus was so contagious though, it made it so that there was a severe lack of testing capability [12].

To rectify this, researchers have been trying to create a substitute model where predictions can be made on the chances a person has contacted COVID-19 or not based on some preliminary symptoms such as fever or dry cough. Though such a model would not completely substitute the accuracy of an in-person test, it can serve as an easily accessible and useful alternative that can help direct medical attention to where it needs to be. Though conducting proper tests might still be required, such a model can serve as an early detection tool and help identify potential patients and target proper medical action towards them.

Once a person is confirmed to have contacted COVID-19, it becomes difficult to assess how their condition will progress. As stated earlier, since it is categorized in a spectrum [11], the medical attention required to each patient can vary immensely based on how the virus chooses to manifest in severity for a specific patient. Thus, with more research conducted on COVID-19, relation between certain symptoms and the chances of a patient's condition worsening to the point of needing extra medical attention can be analyzed. As seen in [3], patients infected with COVID-19 can also have other pre-existing medical conditions that help to deteriorate their condition even further. Reports indicate that people with pre-existing respiratory conditions, heart conditions and diabetes have a much higher chance of developing more severe symptoms. Similarly, data in [11] also shows that the elderly are generally more at risk of developing serious symptoms as opposed to younger age groups, though all age groups do contribute to the death rate. Predicting the chance of person developing such severe symptoms due to their medical history then also becomes a priority as it would help health care workers to divert their attention and resources to people that require it.

Significant efforts are thus being taken to develop such prediction models (based on machine learning), that takes the data of those infected, and their symptoms, and is trained to predict whether their symptoms will develop into more serious conditions requiring extra medical attention or not [11][14][9]. In [14] for example, a machine learning prediction model was set up to predict whether or not the patient in question would require to be transferred to the ICU based on a number of parameters. Similarly, in [5], a machine learning prediction model was developed to predict the mortality risk of patients affected by COVID-19 based on age group, sex and exposure.

The importance of developing such a prediction model is thus paramount, as it would greatly assist health care officials and hospital management in deciding where and to whom they should be dedicating their utmost care and resources. It not only would help the health care system operate much more efficiently, it would also help deliver proper care to those that need it most. Another dimension to consider is the after effects of COVID-19 or more specifically, what long term effects people have after recovering from COVID-19. Heart arrhythmia, brain fog, hypertension and lung damage have all been reported as possible after effects of COVID-19 among recovered patients. Being able to predict such after effects would help health care workers to address those issues very early on and may prevent said conditions from worsening as a result.

1.2 Problem Statement

There are thus three predictions in three different timelines that can be considered to be equally important:

1. The prediction of a person having COVID-19 based on their preliminary symptoms.
2. Given that a person is confirmed to have COVID-19, the prediction that their condition worsens to the point of needing extra medical attention, based on their existing medical history.
3. Given that a person recovers from COVID-19, the prediction of this virus having lasting impact on their health that may require medical attention.

Taking all these dimensions into account, it can be observed that there is significant research being done on each of them and various prediction models being proposed for each if them as well. However, there does not seem to be one singular model that approaches all these dimensions holistically.

Thus, this research aims to answer the following question:

Can a prediction model be developed so that it can predict whether or not a person having COVID-19 based on their symptoms? If they do have COVID-19, can it also predict the chances of their condition worsening based on their pre-existing medical history? Given that they recover from COVID-19; can the model also predict if the virus will leave any lasting medical effects on them?

1.3 Objective and Contributions

This research aims to create a machine learning model for each of the timelines described above. Though such models are already being researched on, such as in [11] and [12], this research also aims to find out a model that can perform well throughout all three timelines. It aims to make use of different machine learning algorithms – namely Random Forest, Naïve Bayes and K-Nearest Neighbors (KNN), and consequently conduct a comparative analysis to find out which one would best serve throughout the three timelines and make the best prediction over the three dimensions.

The objectives of this research are:

1. To create a prediction model(s) for all the three timelines described above.
2. To find the best machine learning algorithm to make predictions over the three timelines.
3. To help gain a better understanding of COVID-19 by inspecting the ways in which it relates to factors such as symptoms, physical characteristics of patients such as age and pre-existing medical conditions.
4. To help health care workers divert extremely limited resources to those that need it most.
5. To help lower the burden on hospitals by efficient management of resources made possible by the prediction model.

1.4 Thesis Structure

Within this paper, we first attempt to establish the foundations of this research. The following section thus contains the literature review and a background on each of the machine learning algorithms we employed in our experimentation, namely Random Forest, Naïve Bayes and K-nearest Neighbor. The aforementioned literature review encompasses all the research papers we found related to our topic in some form or another.

Following this, we have the Proposed Model section, which contains the Dataset Description and Model Description sub-sections. The Dataset Description is further segmented into the Data Pre-processing and Feature selection blocks. As the name implies, the entirety of the Dataset Description subsection is used to describe the dataset we put to use; namely how it was pre-processed from its raw and initial state and how the features were selected before use. The Model Description sub-section is then where we provide a high-level view of our proposed model.

The Experimentation section is next. This is where the paper goes into details of how experimentation was conducted using the different machine learning algorithms. It contains all the necessary steps conducted when employing each algorithm, as well as an overview for how we evaluated the performance of each of them, though this is discussed more extensively in the next section, Results and Analysis. It also contains details on how cases of train-test splits were created on the basis of time and how the same machine learning algorithms employed in the earlier experimentation was employed once more on these new cases of train-test splits and how the performances of each algorithm was evaluated and recorded. Lastly, it goes into the details of how the dataset is split in to three-month time blocks or “time periods” and what features are selected to be analyzed over those time periods. The metrics and reasoning behind choosing each of those features for the additional analysis are also discussed in that part of the section, along with the methods conducting the extra analysis.

In the penultimate section of Results and Analysis, the accuracy scores, confusion matrix, AUROC scores and ROC curves from using each of the machine learning algorithms over the entire dataset is the first thing that is displayed. Following that, the results of the time-split analysis are also shown with tables containing the case of train and test data its corresponding accuracy scores when using each of the machine learning algorithms employed. A visual representation of the comparison of all the algorithms used in these time-split cases is also present. Afterwards, the results of the feature analysis are given. These results are displayed in tabular format across several tables, each serving a specific context of the analysis being conducted. All these results are then visualized in the Feature Analysis Visualization subsection. The Implications subsection that follows describes what each of the results mean within the context of the research being conducted.

Lastly, the Conclusion section looks once more at the goal of this research. It also touches on certain limitations of this particular paper and how it can be expanded upon for future work.

Chapter 2

Related Work

2.1 Literature Review

As stated previously, with there still being many unknown factors to the SARS-CoV-2, significant research is being conducted into it – starting from its more biological characteristics, to the aforementioned development of machine learning models.

In [3], the researchers used three approaches to identify the relationship between COVID-19 and pre-existing medical conditions between patients: a meta-analysis of available retrospective cohort studies of COVID-19 patient data that focused on comorbidity and selected clinical features, aggregating a COVID-19 dataset and identifying important comorbidity associations and finally applying machine learning algorithms to the aggregated data to classify these comorbidities with a mortality rate. A total of 13,400 COVID-19 patients from twenty-six studies were taken for meta-analysis in [3]. Among them, 2,964 patients developed a severe condition or had to be admitted in the ICU or had died. Following the meta-analysis of already published literature, more recent COVID-19 data available from online repositories were assessed as well and used to apply additional machine learning methods that helped compliment the initial meta-analysis. Following that, a machine learning analysis was done on this dataset of 1,143 patients. Six different machine learning algorithms were used: Random Forest, Decision Tree, GBM, XGB, SVM and LBGM.

The most significant comorbidities found through the research in [3] were hypertension, diabetes and metabolic diseases, chronic kidney disease, chronic obstructive pulmonary disease (COPD), asthma and malignancy. Age and sex were however the most significant predictor of COVID-19 mortality as per the paper.

Now in [11], the research was done with data from Brazil and was taken in two sets. It consisted of about 4,826 patients for the training cohort and 3,617 patients for the validation cohort. The researchers employed a few supervised machine learning models to the developed dataset in order to create computational models capable of predicting the disease's outcome. The models used were: Logistic Regression (LR), Linear Discriminant Analysis (LDA), Naïve Bayes (NB), K-Nearest Neighbors (KNN), Decision Trees (DT), XGBOOST (XGB) and Support Vector Machine (SVM). Results from the models used in [11] further show that COVID-19 patients have a greater mortality rate when they are over 60 years old, have respiratory distress and comorbidity such as kidney disease, diabetes, cardiac disease and obesity. Smoking was shown to also increase the chances of death due to COVID-19.

The research proposed in [14] is perhaps most in line with the prediction model we wanted to create for our second timeline where we wanted to predict the chances of a COVID-19 patient developing severe symptoms based on pre-existing medical conditions. It proposes a machine learning based risk-prioritization tool that predicts who will need ICU transfer within 24 hours. Studies indicate that about 20-30% of those infected by COVID-19 require to be hospitalized, with another 5-12% patients from that requiring to be put into the Intensive Care Unit (ICU). With an increase in cases, there might be a much greater number of patients requiring ICU care.

The use of a supervised machine learning model helps to analyze and interpret a patient's clinical and laboratory values and temporal changes. It also helps to put a number to their risk of being in danger of deterioration and hence their need for an ICU transfer. The main goal of the research [14] is to develop a supervised machine learning identifier that predicts the risk of ICU transfer within the next 24 hours for COVID-19 patients using hospital EMR data. Random Forest (RF) approach was used here. The cohort for the study used for the paper [14] was of patients 18 years or older diagnosed with COVID-19 and admitted to Mount Sinai Hospital in general patient beds initially. Though a model was successfully trained and set up for the research, the relatively small dataset and low amount of ICU transfers that resulted from having such a small dataset served as limitations to the research and its results.

Along the lines of the research that dealt with medical resources as that proposed in [14], [9] focused on using machine learning based approaches to predict the probability of when a patient infected by COVID-19 is discharged. In general terms they are focused on how long a COVID-19 infected patient will be under medical care in a hospital. The purpose of this research [4] was to provide hospitals with an estimate of how long their resources are going to be occupied with a specific patient. Multiple machine learning algorithms such IPCRidge, CoxPH, CoxNet, Stagewise GB, Componentwise GB, Fast survival SVM, Fast kernel Survival SVM were used in the paper and performance metric was calculated to compare their reliability. According to that performance metric, they show that for the given features, Stagewise GB proved to be the most accurate at predicting the discharge time of a patient. The researchers in [9] also created two models, both calculating the probability of discharge time; one that predicts discharge time from the moment symptoms start showing and the other that predicts discharge time from the moment the patient has been admitted to the hospital.

For [8], the paper's main objective was predicting mortality among confirmed COVID-19 patients in South Korea. A total of 3022 COVID-19 patients were used along with socio-demographic and exposure as inputs. The final output variable was mortality which gave a binary 'yes' or 'no' to represent if the patient lived or died. Socio-demographic information used in the research [8] included age, sex, province, date of diagnosis and exposure included hospital, religious gatherings, gym facility etc. Five machine learning models were used to predict the outcome and these were Logistic Regression, Support Vector Machine(SVM), K neighbor classification(KNN), Rain Forest(RF), and Gradient Boosting. The dataset used in the research was split into 80 percent for training and 20 percent for testing. The performance of the algorithm was judged based upon discrimination, calibration and overall performance. All the five algorithms used were judged on the basis of metrics of Area under ROC curve, Accuracy, Mathews correlation coefficient and Brier score. The results obtained from the research were: Gradient Boosting performed the best with an accuracy of 0.987 whereas KNN scored the least score (0.979). Gradient Boosting also scored highest on Mathews correlation coefficient but with a lowest score in Brier score. And lastly Gradient Boosting achieved the highest score of 0.966 in AUC in comparison with other algorithms. Overall, Gradient Boosting was the model that performed the best in the paper for predicting accurate mortality risk.

Similarly, in [1], the research focused on three supervised Machine Learning Technique for predicting the survival outcome of COVID-19 patient. The study used information of each patient's demographic, epidemiological, clinical characteristics. This included age, sex, hospital length of stay, travel history, symptoms and whether they survived or died of the disease. Three prediction models were used in the paper: Linear Discriminant Analysis(LDA), Support Vector Machine(SVM) and lastly Random Forest(RF). For measuring the performance, seven evaluation metrics of two different metric (precision, accuracy, sensitivity, specificity, F-score and Kappa) and rank metric(AUC-ROC) were used. Confusion Matrix measured the performance of each models after inputting test data on the learning algorithm and following results were observed in the research [1]: LDA scored 95%, SVM scored 94% and lastly RF which scored the most with an accuracy of 100% in classification (Died, Discharged and Total Column).

[4]focused on the South Korean demographic, with the nationwide cohort of South Korea being used to develop a machine learning model to predict prognosis of COVID-19 based on sociodemographic and medical information. For machine learning, the least absolute shrinkage and selection operator (LASSO), linear support vector machine (SVM), SVM with radial basis function kernel, random forest (RF), and k-nearest neighbors were tested. Here, the models predict the outcome but also early mortality (i.e. 14-30 days). LASSO and linear SVM demonstrated high sensitivities (¿ 90%) in predicting mortality in the paper. The paper [4] also showed that age, sex, moderate or severe disability, the presence of symptoms, and comorbidities including hypertension, DM, chronic lung disease or asthma, and cancer were significant factors in predicting mortality in COVID-19 patients.

In [12], the researchers tried to create a more accurate diagnosis model of COVID-19 than what was available at the time this was written, based on patient symptoms and routine test results. They aimed to do this by applying machine learning to analyze COVID-19 data and get insight into correlation of clinical variables, find subtypes of COVID-19 patients through clustering and create a computational classification model that could discern between patients of COVID-19 and influenza (that showed similar symptoms) based only on clinical variables. For data collection, clinical data was manually collected from PubMed. Following that, correlation between the variables were found. Clustering was also conducted to identify patient sub-types by using Self Organizing Map or SOM. For classification, the XGBoost or eXtreme Gradient Boostine machine learning algorithm was used. The dataset was split into 80%-20% for the training and testing sets respectively. 5-fold cross validation was performed on the data and then the data was fed into a Bayesian optimization function in order to find the best hyperparameters to feed into the XGBoost algorithm. XGBoost was also performed on patient subgroups.

The results of the research conducted in [12] showed that there were several associations present between clinical variables. It was also found that COVID-19 patients could be divided into sub-groups based on certain criteria (serum levels of immune cells, sex and reported symptoms). It was also observed that an XGBoost classification model for separating COVID-19 and influenza patients could be trained to reach a sensitivity of 92.5% and specificity of 97.9%.

Researchers in [13] set up a machine learning model that predicts COVID-19 test on eight features, namely sex, if age is above 60 or not, known contact of infected individual and five clinical symptoms (cogh, fever, sore throat, shortness of breath and headache). The training set used for the machine learning model had records from 51,831 tested individuals, while the test set contained information of 47,401 tested individuals. The model itself was set up with predictions generated by the use of a gradient-boosting machine with decision-tree base learners. By using the gradient-boosting predictor, missing values were inherently handled. The results that followed showed that for the test sets, the model was able to predict with 0.90 auROC with 95% CI. It was also found that fevers and cough were key features that helped predict the disease. Close contact with another individual who contacted the virus was also an important feature.

The paper in [7] focuses on creating a web base platform to predict weather someone should test for COVID-19 given that he or she is exhibiting certain symptoms. The objective of this research is to create a machine learning based web application to predict weather someone is likely to be positive of COVID-19 or not. If they are likely to have it, they are advised to seek medical advice and give an input of their symptoms. The machine learning algorithm they used for this purpose is multinomial Naïve Bayes'. They first used a data set containing 7 attributes, a feature set consisting of 5 symptoms contacts and class all of which are categorical variable. [] They then ran the multinomial Naïve Bayes' algorithm on different ratios of train test split to determine the best ratio to train the model and implement it. They used 4 ratios 70:30, 60:40, 50:50, 40:60. Out of the 4 ratio 50:50 had the best overall performance in weighted performance average along with having the best precision score and the 2nd best recall score behind 70:30 split making it the best split to be selected to train the model for its implementation on the web application.

In [10], a collection of covid-19 patient's data was fetched from Massachusetts General Brigham(MGB) healthcare database to create a model to predict ICU admission within 5 days for a covid-19 patient. 18 machine learning algorithms were employed and evaluated to see which of them performed the best. The data used included demographic data, medical conditions, history of past illness, clinical features, medication used and laboratory findings of COVID-19 patients who were admitted to the Emergency Department (ED) at MGB. The 18 machine learning algorithm resides under 9 broad categories like ensemble, Gaussian process, linear, Naïve Bayes, nearest neighbor, support vector machine, tree-based, discriminant analysis and neural network model. Firstly, the ICU admission prediction models using cross validation displayed all ensemble-based models had mean precision-recall area under curve (PR AUC) of more than 0.77 and Logistic Regression and Linear Discrimination Analysis showed a result of 0.79 and 0.76 respectively. The Logistic Regression also performed similarly but the ensemble algorithm gave the best result of PR AUC and ROC AUC. The results took a different turn when comparing the performance of prediction models based on internal and external validation. In the internal validation dataset, the ensemble methods resulted to PR AUC¿0.8 and LogisticRegression with a PR AUC of 0.83 which gave the best result whereas in the external validation dataset, Bagging Classifier, Random Forest Classifier, Logistic Regression and XGBClassifier resulted better PR AUC than other ensemble models. For better prediction, SHAP analysis was done on individual variables to check the impacts of each variable in the model using random forest.

The research in [5] emphasized on how to predict risks for critical patients of COVID-19 based on baseline clinical parameters. The main aim of the paper was to predict the critical condition of patients after being admitted to the hospital. The researchers defined critical conditions where patient may require machine ventilation, multi-organ failure, needs admission in ICU and death. The study was conducted at a hospital named Sheba Medical Center between March till April of 2020 where 162 patients had confirmed COVID-19 infection. Three different machine learning algorithms Artificial Neural Network (ANN), Random Forest (RF) and Classification and Regression Decision Tree (CRT) were applied. The results showed that the Artificial Neural Network (ANN) came up with ROC AUC result of 0.92, Classification and Regression Decision Tree (CRT) with ROC AUC result of 0.90 and Random Forest (RF) algorithm gave the best ROC AUC result of 0.93 among the other algorithms.

Next, the researchers in [2] used Extreme Gradient Boosting (XGBoost) to predict hospital mortality and critical events at time windows of 3,5,7 and 10 days from patient admission. The population of the study included five hospitals in New York for 4098 Covid-19 patients.

All the models used in the paper were validated using 10-fold stratified cross-validation and confidence intervals were generated using 500 iterations of bootstrapping. XGBoost performed well for critical event prediction, with an AUC-ROC of 0.80 at 3 days, 0.79 at 5 days, 0.80 at 7 days, and 0.81 at 10 days. Mean absolute SHAP values were calculated for each XGBoost model in the internal validation data set. Notable drivers of predictability included both systolic and diastolic blood pressure, pH, total protein levels, C-reactive protein, and D-dimer. For mortality, both high and low values for age, anion gap, C-reactive protein, and LDH were the strongest effectors in guiding mortality prediction within one week of admission.

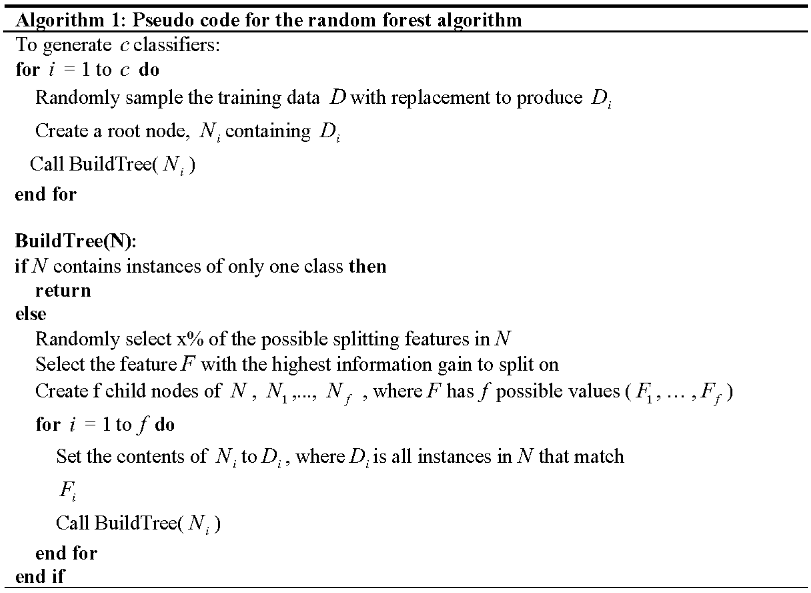
Their results did however have certain limitations such as their predictions relying solely on data extracted around patient admission (ie, within 36 hours).

In the final paper in [6], the research conducted was a little different as the researchers developed a machine learning model using five serum chemistry laboratory parameters. They presented a review study that assessed research center information and mortality from patients with positive RT-PCR examine results for SARS-CoV-2. The goal of this investigation was to recognize prognostic serum biomarkers in patients that were at most serious danger of mortality. They built a machine learning model utilizing five serum chemistry laboratory parameters (c-responsive protein, blood urea nitrogen, serum calcium, serum egg whites, and lactic acids) from 398 patients (43 terminated and 355 non-lapsed) for the forecast of death up to 48 hours preceding patient termination. The subsequent help vector machine model accomplished 91% sensitivity and 91% specificity (AUC 0.93) for foreseeing persistent lapse status on held-out testing information.

2.2 Machine Learning Algorithms

2.2.1 Random Forest

Random Forest is a supervised machine learning algorithm that makes use of decision trees. Decision trees are tree-shaped structures that are used to determine the course of an action. The structure is designed such that there is a root node, decision nodes, leaf nodes and branches. The principle a decision tree works on is that of entropy. Entropy is defined as the measure of randomness or unpredictability of a dataset. At each step, the dataset is split (from the root node) into the leaf nodes based on some attribute or another. At each split, there is an information gain, or a measure of the decrease in entropy relative to the previous step. The idea the tree works on is that each split is done in such a way that the information gain is the highest, resulting in maximum decrease in entropy.



```
Algorithm 1: Pseudo code for the random forest algorithm
To generate $c$ classifiers:
for $i$ = 1 to $c$ do
    Randomly sample the training data $D$ with replacement to produce $D_i$
    Create a root node, $N_i$ containing $D_i$
    Call BuildTree( $N_i$ )
end for

BuildTree(N):
if $N$ contains instances of only one class then
    return
else
    Randomly select x% of the possible splitting features in $N$
    Select the feature $F$ with the highest information gain to split on
    Create f child nodes of $N$ , $N_1$ ,..., $N_f$ , where $F$ has $f$ possible values ( $F_1$, … , $F_f$ )
    for $i$ = 1 to $f$ do
        Set the contents of $N_i$ to $D_i$ , where $D_i$ is all instances in $N$ that match
        $F_i$
        Call BuildTree( $N_i$ )
    end for
end if
```

Figure 2.1: Pseudocode for Random Forest algorithm

A Random Forest classifier makes use of several of these decision trees. It works by building multiple decision trees and then merges them together to make a prediction. The results obtained from each individual tree is then averaged together to produce the actual results of the Random Forest. Essentially the results work similar to a voting or election. The majority "vote" is taken, or in this case, the majority decision output is taken. The pseudocode for the Random Forest algorithm is given above in Figure 2.1.

When using the Random Forest classifier using the sklearn library, it allows the use of multiple hyperparameters. For our paper, we used two of them: "n_jobs" and "random_state". These both make the model work faster. The n_jobs parameter is used to tell the engine the number of processors it can use to run the algorithm. We set it to two, meaning that we are allowing it to use two processors for computation. The random state parameter is used to ensure that the model can produce the same results once given the same hyperparameters and same training data.

2.2.2 K-nearest Neighbors

KNN algorithm is an algorithm used in supervised Machine Learning to serve regression and classification problems [16]. Dealing with classification problems, the predicted final output is always a discrete value, either yes/no, favor/disfavor, binary 1/0 etc. This algorithm takes K inside the parameter which denotes the number of nearest neighbor data values. The KNN classification model completely relies upon the number of nearest neighbor data values present in the dataset.

This algorithm uses distance function to predict the class of any new instance. Its distance function includes Euclidean distance, Manhattan distance and Minkowski distance which are used for continuous variable. The decision of any new instance is calculated with the help of distance function and based on the k value, the category of the new instance is decided. Selection of the k value has no universal limitation but k must be greater than equal to 1. However, altering the k value in the model and running on that specific dataset may help to find the best k value for which the algorithm accurately predicts the class from the dataset. In our model, since we are working with categorical variables, Hamming distance function is used which is specifically built for discrete variables [15].

2.2.3 Naïve Bayes

Naïve Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. Bayes' Theorem States that if there to predictors "B" and "C" used to predict the outcome component A then

$$\mathbf{P\ (A|C,\ B) = P\ (C,\ B|A)\ {*}P(A)/P\ (C,\ B)}$$

Naïve Bayes' on the other hand relaxes this assumption stating the relationship to be

$$\mathbf{P\ (A|C,\ P) = P\ (B|A)\ {*}P(C|A)\ {*}B(A)}$$

Naïve Bayes assumption of predictors being independent makes it much faster and easier to train this also helps it predict test sets faster. And if the give assumption of predictors being independent hold then Naïve Bayes' out performs most machine learning algorithms.

On the other hand, the assumption of independent predictors is a problem in real life as in most cases variables are not independent. Also, if is any one of the categorical variables in the prediction data does not have an observation that is present in the testing data then the probability for that observation will be 0 and the model will not be able to make any prediction. This is call zero frequency but it can be solved by using smoothing technique for example Laplace smoothing.

For Timeline 2, we used three variants of Naïve Bayes - Gaussian Naïve Bayes - Multinomial Naïve Bayes and - Bernoulli Naïve Bayes. Gaussian Naïve Bayes is like general Naïve Bayes' but it allows features to be continuous and assumes that they follow normal distribution. Whereas Multinomial Naïve Bayes' is used on multinomially distributed data but the features have to be discrete. Lastly Bernoulli Naïve Bayes' assume all the feature in the data to be binary values.

Chapter 3

Proposed Model

3.1 Dataset Description

Finding data related to COVID-19 is both difficult and easy depending on what kind of data one is looking for. For our research, we needed data detailing individual symptoms (for Timeline 1) and pre-existing medical conditions (for Timeline 2).

We also needed data that contained information about a person's health condition post COVID-19, however we are still searching for such a dataset as it could not be immediately found. Similarly, the dataset for Timeline 1 was also difficult to find as it contained private information about the patients. We have contacted a few researchers who have written papers that used such data in an attempt to get usable datasets for Timeline 1, however the data could not be procured in time for this paper's submission deadline. For future expansion of this paper, we are optimistic about our chance of finding data for those time lines.

Timeline 2 data was also difficult to come-by due to private nature of information required. Due to being in quarantine for the pandemic, it was also difficult to conduct the data collection process ourselves from local hospitals. Thus, we had to rely on online repositories and other researchers for our data. Hospital records and any other kind of data available for COVID-19 patients usually just indicated the patient's sex, COVID-19 test results and sometimes, their region of residence. They rarely included personal details such as symptoms and/or medical history containing their pre-existing medical conditions. However, one such dataset was procured from the Government of Mexico's official website that contained necessary information, making it usable for our proposed model's Timeline 2. The dataset contained over information from over 10,000,000 people affected by COVID-19 from Mexico. It is available for free and the raw dataset can be downloaded from the aforementioned Mexico government website.

Though the dataset contained enough information for us to use it for Timeline 2 of our model, it was in Spanish (the columns/features) and still contained certain "impurities," such as null values and categorical data. For that pre-processing was required in order to make the data usable for our purposes.

3.1.1 Data Pre-processing

After the initial data for Timeline 2 was collected, it contained data of more than 10,000,000 patients, taken over a period from January 2020 to April 24, 2021, before preprocessing. As mentioned earlier, the columns were initially in Spanish. Thus, the columns had to first be translated to English. The dataset was also designed such that values of 1 meant a binary “Yes” and values of 2 meant a binary “No”. On the sex column, 1 referred to Female and 2 referred to Male. Other values of 97, 98 and 99 were used to describe null or missing values. Similarly, any date value that was given as 9999-99-99 was also used to signify missing date values.

After translation, it was noticed that this dataset had a few columns that were irrelevant to our model, along with cases of null values and categorical data. Having null values and categorical data in our dataset would pose problems for the machine learning algorithms we chose to employ and thus preprocessing had to be conducted on the dataset.

Afterwards, any row that had icu values equivalent to 97, 98 or 99 was removed. This was done because the icu would serve as our target variable. As such, it being missing would be detrimental for the machine learning algorithms. Following this, all values of 97, 98 and 99 for all other features was replaced with a value of 0 to signify the missing values. The admission_date column values was then changed into the datetime format in python (for the time splitting that will be done later). Finally, the age values are grouped in to 4 values, each to represent a certain age group.

The grouping is done as follows:

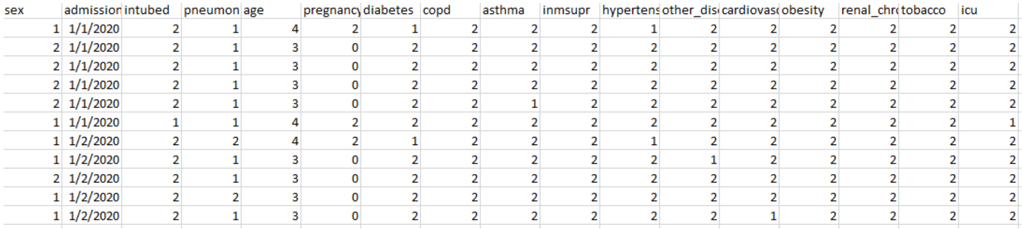
Table 3.1: Age grouping done during data pre-processing

Age Group	**Associated Number**
0 - 18	1
19 - 39	2
40 - 60	3
61+	4

3.1.2 Feature Selection

The columns of id_registration, origin, sector, entity_um, entity_nac, entity_res, municipality_res, patient_type, symptom_date, date_def, nationality, spoken_lang_indig, indigenous, other_case, lab_result, final_classification, migrant, country_nationality, country_origin, date_update, take_sample_lab, take_antigen_sample and antigen_result were all dropped. The purpose of the prediction model for Timeline 2 data is to predict whether a person needs extra medical attention in the future due to their condition worsening or not. This prediction would need to be made on the basis of pre-existing medical condition or physical characteristic of the patient. Any feature that did not fall into those two categories was hence dropped. One exception to this rule was the admission_date feature. This was only kept in order to perform the time-wise train-test split in a latter part of the experimentation. After conducting that split, the column was dropped as well. The resultant dataset thus the following features: sex, admission_date (to be dropped later), intubed, pneumonia, age, pregnancy, diabetes, copd, asthma, inmsupr, hypertension, other_diseases, cardiovascular, obesity, renal_chronic, tobacco and icu.

The dataset after pre-processing is done on it can be seen in Figure 3.1 below.



sex	admission	intubed	pneumon	age	pregnancy	diabetes	copd	asthma	inmsupr	hypertens	other_dis	cardiovas	obesity	renal_chr	tobacco	icu
1	1/1/2020	2	1	4	2	1	2	2	2	1	2	2	2	2	2	2
2	1/1/2020	2	1	3	0	2	2	2	2	2	2	2	2	2	2	2
2	1/1/2020	2	1	3	0	2	2	2	2	2	2	2	2	2	2	2
2	1/1/2020	2	1	3	0	2	2	2	2	2	2	2	2	2	2	2
2	1/1/2020	2	1	3	0	2	2	1	2	2	2	2	2	2	2	2
1	1/1/2020	1	1	4	2	2	2	2	2	2	2	2	2	2	2	1
1	1/2/2020	2	2	4	2	1	2	2	2	1	2	2	2	2	2	2
1	1/2/2020	2	1	3	0	2	2	2	2	2	1	2	2	2	2	2
2	1/2/2020	2	1	3	0	2	2	2	2	2	2	2	2	2	2	2
1	1/2/2020	2	2	3	0	2	2	2	2	2	2	2	2	2	2	2
1	1/2/2020	2	1	3	0	2	2	2	2	2	2	1	2	2	2	2

Figure 3.1: Table Timeline 2 Dataset as a dataframe as seen after preprocessing

The features that remained after pre-processing were all necessary parameters as they all served to either describe the patient's physical characteristics, or describe some relevant pre-existing medical condition that could affect whether the patient required extra medical attention or not. The columns of sex and age were used to describe the physical characteristics of the patients. Following this were columns of potential preexisting medical conditions that the patients might have had. This included columns for pneumonia, pregnancy, diabetes, copd (chronic obtrusive pulmonary disease), asthma, inmsupr (immunosuppressed), hypertension and so on. It also had columns for recording if the person needed to be intubated or not (intube) and finally, the icu column describing if the person needed to be transferred to the ICU. The icu feature was used as the target variable, as it indicated whether a person needed to be admitted to the ICU or not. Since admittance to the ICU infers that a patient's condition worsens to the point of needing extra medical attention, it was the perfect choice for a target variable given the context of the prediction model. There was also an "other diseases" column for any medical condition not accounted for in the other columns. As stated previously, for each column, 1 values indicate a binary "Yes," 2 values indicate a binary "No" and finally 0 values indicate null or missing values (except in case of sex, where 1 meant female and 2 meant male).

3.2 Model Description

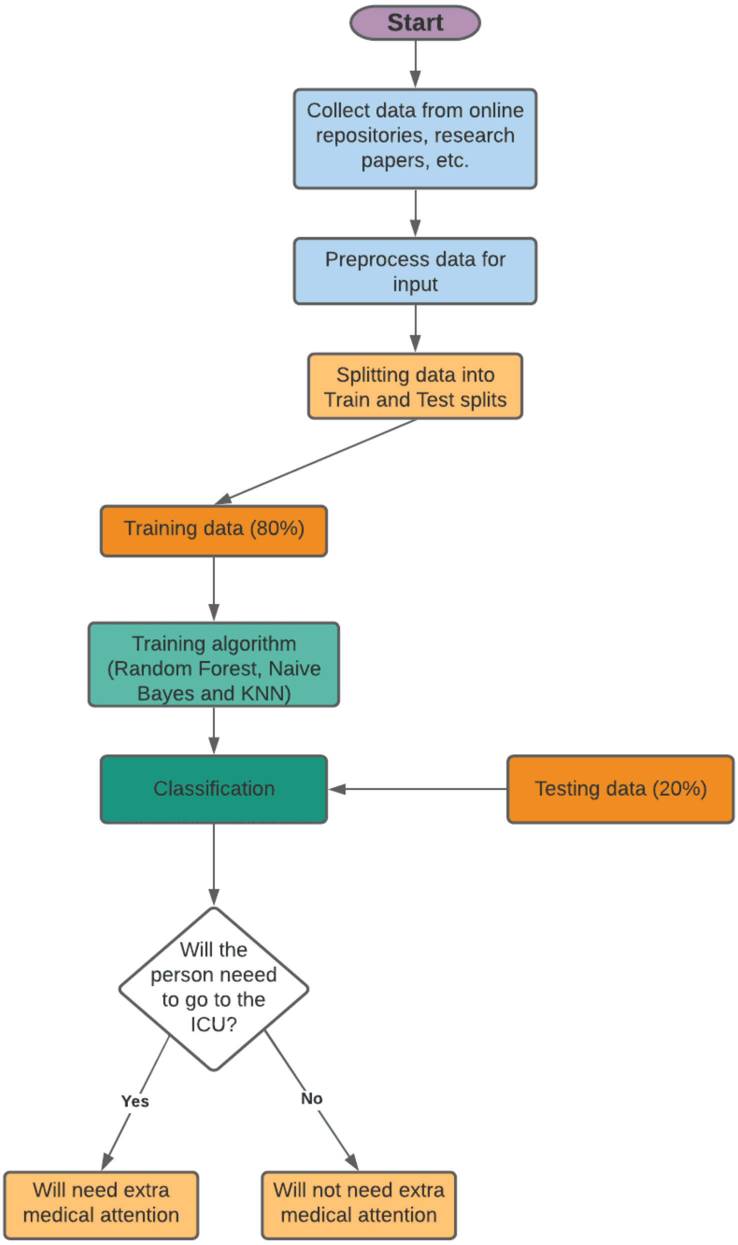

Start
Collect data from online repositories, research papers, etc.
Preprocess data for input
Splitting data into Train and Test splits
Training data (80%)
Training algorithm (Random Forest, Naive Bayes and KNN)
Classification
Testing data (20%)
Will the person neeed to go to the ICU?
Yes
No
Will need extra medical attention
Will not need extra medical attention

Figure 3.2: Flow chart of the proposed prediction model

This proposed model will be used to make predictions on the three different timelines on three different dimensions, as stated previously. To make these predictions, the model is divided into three different parts and trained on three different datasets that have the necessary features. The three parts or Timelines are as stated before:

1. Predicting whether person has COVID-19 based on symptoms

2. Predicting if person's condition will worsen, given he or she is confirmed to have COVID-19
3. Predicting if person who recovered from COVID-19 will have lasting health problems

For each of these parts, the following steps were employed. First, data was collected from online open repositories and research papers. This data was then analyzed and preprocessed so as to only include features that were relevant to our model. Afterwards, the data is split into training and test splits.

The training split is used to train our selected algorithms over the data, and then the test split is used to check the accuracy of the trained algorithm. The classification performed gives two results of either requiring extra medical attention or not. The "extra medical attention" here varies depending on the timeline for which the result is found. For example, for Timeline-1, extra medical attention would mean the person might have COVID-19 based on their symptoms and should seek out medical attention. For Timeline-2, it might mean that the person's condition might severely worsen given their medical history and would hence require extra care. A high level view of the model design is given below. The following steps are performed for all three timelines and their respective dimensions. A high level view of the model design can be observed in Figure 3.2.

Chapter 4

Experimentation

This section describes the implementation of our proposed model. The model was implemented and tested by using python code (python3), through Google colab. The implementation was done in four stages. First, we preprocessed the dataset received from the Mexico government website and preprocessed it according to our needs. The preprocessing details were discussed in section 3 and further details about the dataset are given below in the "Input Data Preprocessing" section. Following this we split the data into training and splitting sets. We accomplished the splitting by implementing the train_test_split library function from the sklearn python library. The split was 80-20, with 80% data reserved for the training set and 20% reserved for the testing set. After this, we finally got to our classification stage by training our machine learning algorithms with the training set of the data, with the icu column set as our y (target variable), and the rest of the columns set as the x (parameters). Finally, once our algorithms are fitted over the training set, we tested them using the testing set. We used three algorithms as previously stated – Random Forest (RF), Naïve Bayes (NB) and K-nearest Neighbors (KNN). To gauge the accuracy of the model and respective algorithm used, we calculated the accuracy by implementing the accuracy_score function from the sklearn library and also displayed the corresponding Confusion Matrix.

The accuracy score calculated by implementing the function is calculated by the formula

$$Accuracy = \frac{True\ Positive + True\ Negative}{N\ (total\ number\ of\ values)}$$

This true positive and true negative values can be observed in the Confusion Matrix, along with the false positive and false negative values

The ROC curve and area under ROC curve (AUROC) was also computed. The ROC curve summarizes prediction model of a classification model by plotting the False Positive Rate (FPR) on the x-axis and True Positive Rate (TPR) on the y-axis. The equations for FPR and TPR are as follows:

$$TPR\ (Sensitivity) = \frac{TP}{TP + FN}$$

$$FPR\ (1 - Specificity) = \frac{FP}{TN + FP}$$

After dealing the comparison of the machine learning algorithms over the entire dataset, there was a series of experiments conducted on different train-test splits of the same data. This splitting of the dataset was done by the use of the admission_date feature. Instead of calling the split function and splitting randomly in a ratio of 80-20 (with 80% data for training and 20% data for testing), the data was split on the basis of time, in cascading amounts, for each "case". After each new case of train-test data was achieved, all three machine learning algorithms were once again trained using each new train-split and run over each new test-split. The corresponding accuracy scores for each of these machine learning algorithms over each of these new cases was also then found.

Once the time-split analysis was conducted, the entire dataset was once again divided on the basis of time, but in a slightly different manner. Now, instead of dividing in cascading amounts for train-test splits, the data was simply segmented into 3 month portions, or 3-month "time periods". This was done in order to get a more detailed insight into how specific features of the dataset interact with the target variable. Once the data segmentation was successful, the number of rows where a select number of features returned positive or "Yes" values were counted and recorded. These select features were namely sex, age (group) and two of the features that showed the highest correlation to the target variable. The last two features were selected after using the corr() function over the dataset and selecting the ones that showed the highest correlation to the icu parameter.

To visually display the results obtained from both the time-split experimentation and the feature analysis for each time period, bar charts were constructed by using the matplotlib library in python.

All other code was implemented and tested using python code as well, through Google colab as stated earlier. Google colab comes with all python libraries available through the cloud, so implementing and testing such machine learning algorithms through the platform made it easier for us as all the python packages required were readily available. Python is especially useful for machine learning algorithms as it, as a language, allows direct interaction with the code. Machine learning is an iterative process where data drives the analysis and thus, it is imperative for such processes to have tools that make quick iteration and easy interaction possible, as is the case with python [14].

4.1 Machine Learning Algorithm Experimentation

All implementation for Random Forest (RF) was done using python and through the Google colab notebook. First the dataset was loaded as a pandas dataframe. Following this, the dataset was split into 80% training set data and 20% testing set data. This, as mentioned previously, was done through the train_test_split function from the sklearn python library. Afterwards, the RandomForestClassifier was imported from the sklearn.ensemble library. This classifier was then fitted to the training set data. Finally, the accuracy of the model with Random Forest was calculated by implementing the accuracy_score function. Additionally, the Confusion

Matrix was also found using the sklearn.metrics library's confusion_matrix function.

The implementation for K-nearest Neighbors (KNN) were used just as with RF, implementation was done with python through the Google colab notebook. The initial steps for the implementation are just as seen in the RF implementation, with the dataset being imported as a pandas dataframe, and the 80-20 splitting of the training and testing set data respectively. The only difference is that instead of importing the RandomForestClassifier, the KNeighborsClassifier is imported from the sklearn.neighbors library. The follow-up steps are similar to the RF implementation as well, with accuracy score being calculated by the accuracy_score function and a Confusion Matrix generated using the confusion_matrix function.

Three variation of Naïve Bayes were used for this paper were used just as the last two, all implementation was done using python and through the Google colab notebook. The first steps of implementation are the same as the implementation of RF and KNN, with the dataset being loaded on as a pandas dataframe. Additionally, all the features in the data frame were printed and a heat map of the feature was generated using seaborn library to visualize the correlation of the features and the features that were not required were dropped. Following this, the dataset was split into 80% training set data and 20% testing set data, as done in for the other two algorithms, through the train_test_split function from the sklearn python library. Afterwards, the Naïve Bayes variations were implemented by first importing the BernoulliNB, MultinomialNB and GaussianNB from the sklearn naive_bayes library. These classifiers were then fitted to the training set data. Finally, the accuracy of the models with the three Navie Bayes's classifiers were calculated by implementing the accuracy_score function, along with the Confusion Matrix using the sklearn.metrics library's confusion_matrix function.

4.2 Time Split Implementation

Though the target of this research was mainly to identify which algorithm performs better in terms of accuracy on predicting whether a patient will require an ICU bed or not with the available medical records, there were certain variables that were not accounted for – least of which was the mutating nature of the COVID-19 virus. In order to see whether the proposed model still performed even for mutated strains of the COVID-19 virus, a few changes were made with the experimentation procedure.

The dataset used in the research includes data from January 2020 till April 24, 2021 consisting of 724,648 number of rows and 17 columns upon which the admission date of each patient is recorded under 'admission_date'. Though initially a random split was incurred with an 80-20 ratio, now an alternative splitting of the dataset in to train and test splits was done on the basis of the aforementioned 'admission_date' column. The admission date feature here was used as an indicator of when the patient was admitted into the hospital and thus, by extension, when the patient contacted COVID-19. Specifically, the entire dataset was then split into time splits, with the train split being incremented by adding 3-months of data for each "case" and testing over the remaining data. The time split are as follows:

- Time Split 1: From January 1, 2020, to March 31, 2020
- Time Split 2: From April 1, 2020, to June 30, 2020
- Time Split 3: From July 1, 2020, to September 30, 2020
- Time Split 4: From October 1, 2020 to December 31, 2020
- Time Split 5: From January 1, 2021 to March 31, 2021

This splitting was done directly in code by the use of python. The dataset was imported into Google colab and a split_date variable was taken as the end point for each split. Then, on the basis of this split_date, the train test splits are accordingly arranged. After the data was divided into the splits described above, alternating train-test splits were used to train the machine learning model and test it. The train-test splits used were as follows:

- Case 1: Train data = Time Spilt 1 (data of the first 3 months)

  Test data = Remaining data (all the data after the first 3 months)
- Case 2: Train data = Time Split 1 and 2 (data of the first 6 months)

  Test data = Remaining data (all the data after the first 6 months)
- Case 3: Train data = rime Split 1, 2 and 3 (data of the first 9 months)

  Test data = Training data (all the data after the first 9 months)
- Case 4: Train data = Time Split 1, 2, 3 and 4 (data of the first 12 months)

  Test data = Training data (all the data after the first 12 months)
- Case 5: Train data = Time Split 1, 2, 3, 4 and 5 (data of the first 15 months)

  Test data = Remaining data (ale the data after the first 15 months)

For each of these cases, the machine learning algorithms described above are once again fitted over the train set and their accuracy over each case is measured over the test set using the same accuracy_score function as described above.

4.3 Time Period Feature Analysis

For deeper data-analysis using the Timeline 2 dataset, instead of separating data on the basis of train test splits, the entire dataset is segmented into 3-month time periods. This was done in order to inspect certain specific features and the trends in patients associated with those features over the chosen time period. The time periods are described as follows:

- Time Period 1: January, 2020, to March, 2020
- Time Period 2: April, 2020, to June, 2020
- Time Period 3: July, 2020, to September, 2020
- Time Period 4: October, 2020, to December, 2020
- Time Period 5: January, 2021, to March, 2021

Although it would be possible to look deeper into all the features (columns) of data available, the goal of this part of the research was to analyze the dataset in a manner that provided insight into the nature of the virus and the way it affects patients individually. In order to accomplish this, certain specific features that could help with a more meaningful representation of trends for patients was chosen. This was done in two ways - first, features that described the patient's innate physical characteristics where chosen. This was namely sex (whether the patient was male or female) and age (which age group the patient belonged to). These features were chosen in order to compare and contrast, based on the data at hand, which subgroups (based on sex or age) were more prone to contracting COVID-19 and needing ICU support once they were infected with the virus.

Secondly, features that showed the highest correlation to the target variable of icu were chosen. These were the features that could most likely be the primary reason propelling a patient towards needing ICU support and thus, seeing how often a patient adheres to each of these features and needs ICU support and comparing them could help provide a clearer picture of just how deeply the correlation between these attributes and a patient's condition worsening is.

The features of age and sex were not dependent on factors such as correlation and so it was possible to readily chose them. To then find the features that have the highest correlation with the target variable, the corr() function was called over the pandas dataframe containing the dataset. This function was called to find the pairwise correlation of the columns in the dataset. The results upon calling it showed that the features with the highest correlation were intubed (with a correlation score of 0.352) and pneumonia (with a correlation score of 0.134). The features of obesity, pregnancy and inmsupr had the next highest correlation (each having a correlation score of 0.0425, 0.0246 and 0.0206 respectively).

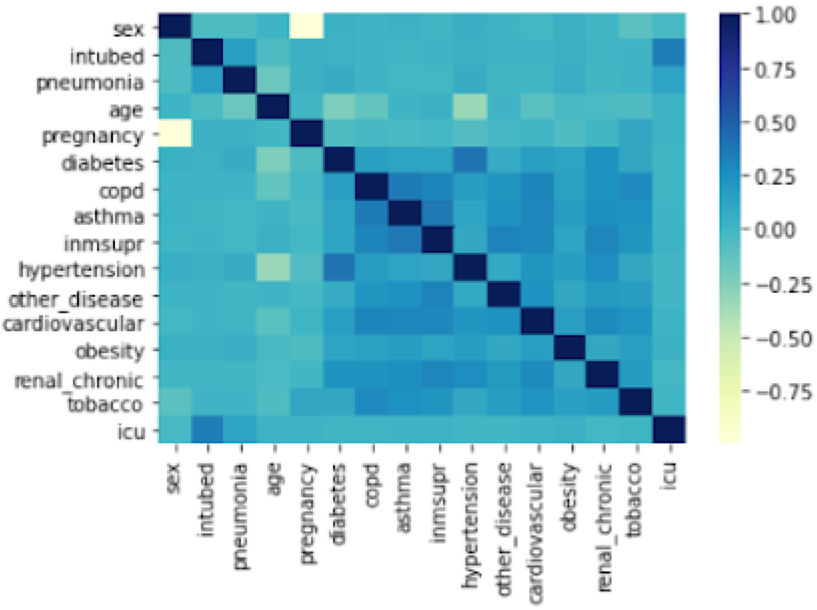


Figure 4.1: Seaborn heatmap of features

A visual representation of the correlation (found using the seaborn heatmap library) can be seen in Figure 4.1 above. This was done by importing the seaborn data visualization library and then using it over the data correlation. With these results, the features of intubed and pneumonia were the ones selected on the basis of correlation. The features of obesity, pregnancy and inmsupr, which also showed relatively high correlation were also noted, in order to perform comparative analysis with the two features chosen that showed the highest correlation.

With that, the features over which additional analysis would be conducted were: sex, age, intubed and pneumonia. The total number of COVID-19 patients that adhere to each of these selected features for each time period was first found. This was done by simply calling the len() function over the dataframe for rows that meet desired conditions (for example, for rows where age = 1). Similarly, the total number of patients that adhered to these features and also needed ICU support was also found.

Once these values were obtained, the results were visualized using bar charts. For the features that represent a patient's physical characteristics (i.e. sex and age), individual bar charts were generated for each of the two features. The bar chart generated for sex, showed the total number of male and female patients of COVID-19 for each time period. Another rendition of this bar chart - this time showing which male or female patients also needed ICU support for each time period - is also generated. Similarly, two renditions of bar charts for age are also generated. The first showed the total number of patients that belonged to age group and how that varied over each time period. The second, similarly, showed the number of patients that belonged to each age group and also needed ICU support and how that changed over the time periods.

Representing the high correlation features in a meaningful manner required the help of a few more features - namely, obesity, pregnancy and inmsupr (which were the next highest correlation features). As stated previously, obesity, pregnancy and inmsupr were also chosen for comparative analysis, but the analysis done on them was limited in comparison to that done on the other features listed previously. This was because the main purpose of including these three features was to compare and contrast against the intubed and pneumonia features. Instead of constructing the bar charts for the total number of patients that adhered to each feature as before, it was decided that a more meaningful representation of these high correlation features' data would be through finding the percentage of patients that adhered to each feature and needed ICU support out of all patients adhering to that feature over the time periods. This was done by the use of the following formula:

$$\frac{No.\ of\ Patients\ that\ adhere\ to\ the\ selected\ feature\ AND\ need\ ICU\ support\ in\ that\ time\ period}{No.\ of\ Patients\ that\ adhere\ to\ the\ selected\ feature\ in\ that\ time\ period}\ X\ 100\%$$

This formula would essentially give the percentage of patients associated to a specific feature, who ended up needing ICU support out of all patients adhering to that feature (for that time period). For example, for intubated icu time period 2, the formula would be:

$$\frac{No.\ of\ Patients\ that\ had\ to\ be\ intubated\ AND\ need\ ICU\ support\ in\ time\ period\ 2}{No.\ of\ Patients\ that\ need\ to\ be\ intubated\ in\ time\ period\ 2}\ X\ 100\%$$

The result obtained from the equation above would reflect the percentage of people who were intubated that also needed to get admitted into the ICU (in time period 2).

Thus, using this formula, the percentage of patients for each of the 5 features of intubated, pneumonia, obesity, inmsupr and pregnancy were found, for each time period. The values obtained over each of these time periods where then visualized over a single bar chart.

Chapter 5

Results and Analysis

5.1 Accuracy Scores and Confusion Matrix

After running the algorithms for our proposed models, the results we obtained can be summarized as follows:

First, with the Random Forest (RF) classifier running on test data consisting of 144930 rows, we achieved an accuracy score of 0.92(5). The Confusion Matrix achieved using the RF classifier is given in the Table 5.1 below.

Table 5.1: Confusion Matrix when using the RF Classifier

N = 144930	**Predicted: NO**	**Predicted: YES**
Actual: NO	*(True Negative)* 552	*(False Positive)* 507
Actual: YES	*(False Negative)* 10241	*(True Positive)* 133630

Next, with the K-nearest Neighbor (KNN) classifier running on the same test data consisting of 18647 rows, it also achieved an accuracy score of 0.92(1). The Confusion Matrix achieved with KNN is given over below in Table 5.2.

Table 5.2: Confusion Matrix when using KNN Classifier

N = 144930	**Predicted: NO**	**Predicted: YES**
Actual: NO	*(True Negative)* 1160	*(False Positive)* 9633
Actual: YES	*(False Negative)* 1802	*(True Positive)* 132335

Finally, with the Naïve Bayes(NB) classifier running over the test data, an accuracy score of 0.92 (5.3) was achieved with the Multinomial variation. Similarly, the Bernoulli variation of NB yielded an accuracy score of 0.92 (5.4) and the Gaussian variation gave an accuracy score of 0.86 (5.5). The Confusion Matrix achieved from running the all three NB classifiers are given below in Tables 5.3, 5.4 and 5.5.

Table 5.3: Confusion Matrix when using the Multinomial NB Classifier

N = 144930	**Predicted: NO**	**Predicted: YES**
Actual: NO	*(True Negative)* 0	*(False Positive)* 0
Actual: YES	*(False Negative)* 10665	*(True Positive)* 134265

Table 5.4: Confusion Matrix when using the Bernoulli NB Classifier

N = 144930	**Predicted: NO**	**Predicted: YES**
Actual: NO	*(True Negative)* 99	*(False Positive)* 506
Actual: YES	*(False Negative)* 10566	*(True Positive)* 133759

Table 5.5: Confusion Matrix when using the Gaussian NB classifier

N = 144930	**Predicted: NO**	**Predicted: YES**
Actual: NO	*(True Negative)* 5362	*(False Positive)* 14390
Actual: YES	*(False Negative)* 5303	*(True Positive)* 119875

5.2 ROC curves and AUROC scores

For each of our proposed algorithms, we found their ROC curves and corresponding AUROC or area under ROC curve scores. The results are given below:

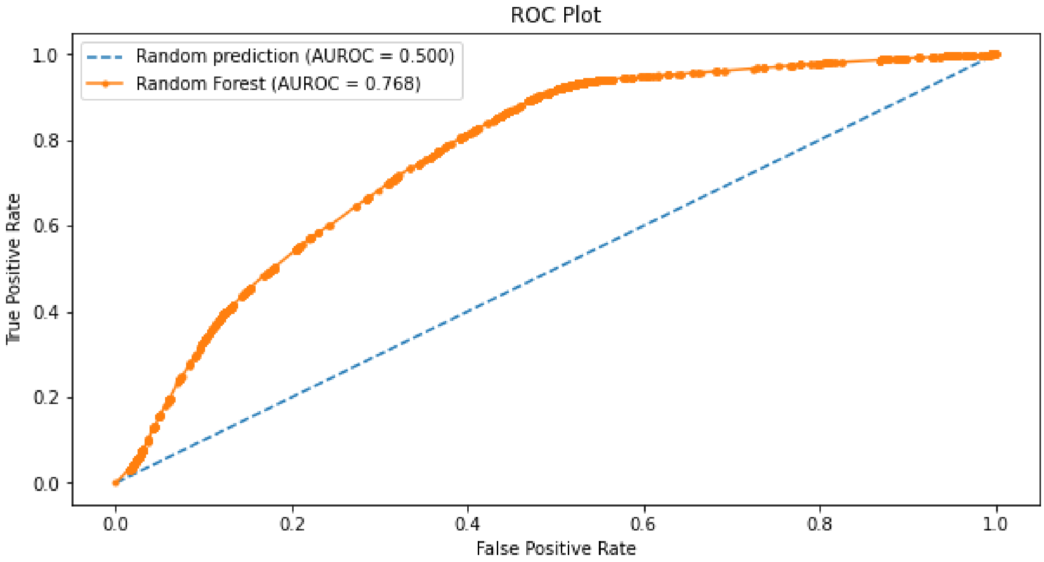


Figure 5.1: ROC curve generated using Random Forest Classifier

Using the Random Forest Classifier generated an AUROC score of 0.768. The corresponding ROC curve and a random prediction curve (for reference) can be seen above in Figure 5.1.

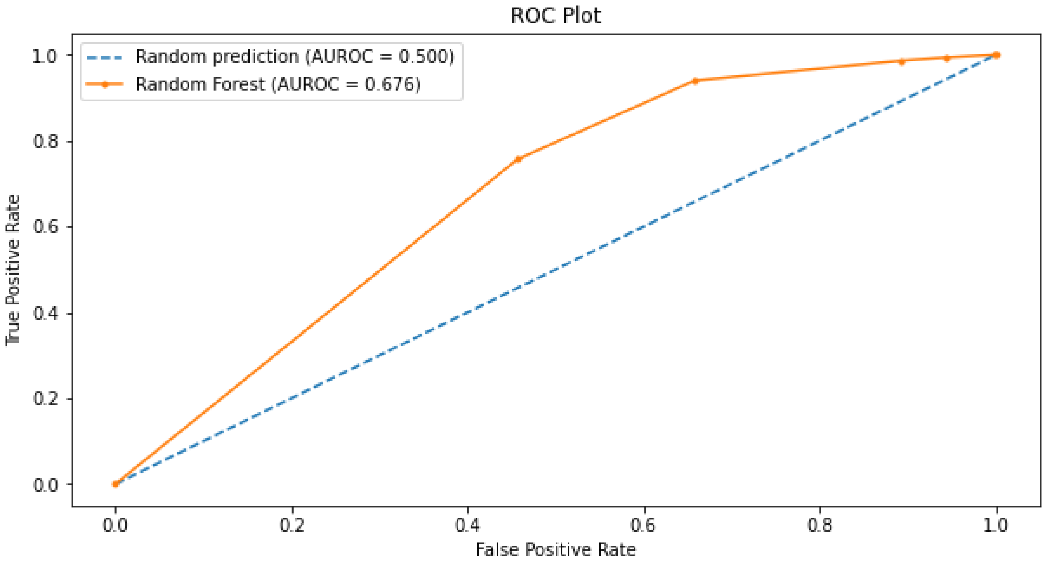


Figure 5.2: ROC curve generated using K-nearest Neighbors Classifier

With the K-nearest neighbors or KNN classifier, an AUROC score of 0.676 was achieved. Figure 5.2 above shows the ROC curve generated using KNN, along with another curve of random probability.

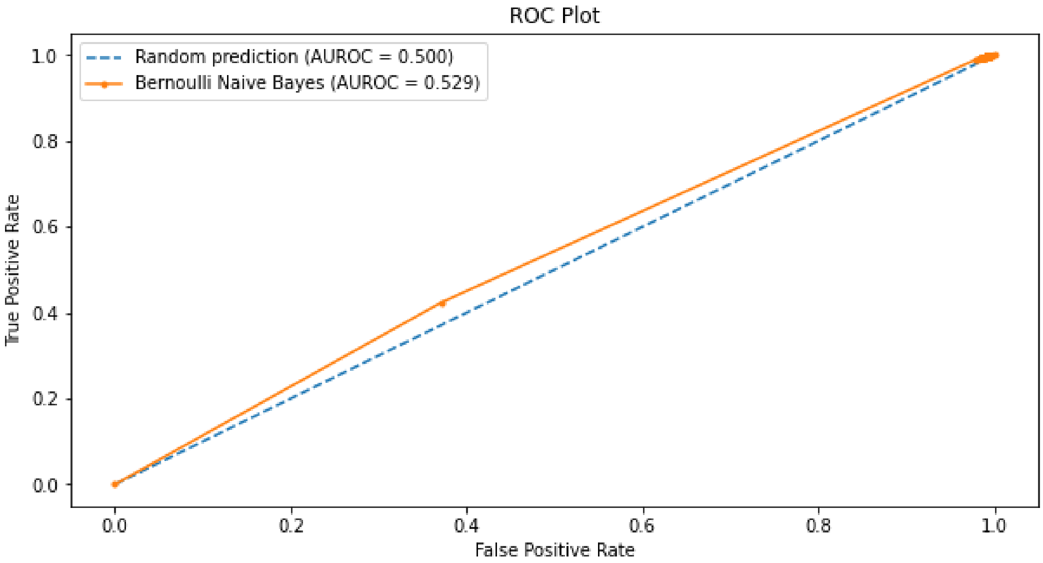


Figure 5.3: ROC curve generated using Bernoulli Naive Bayes

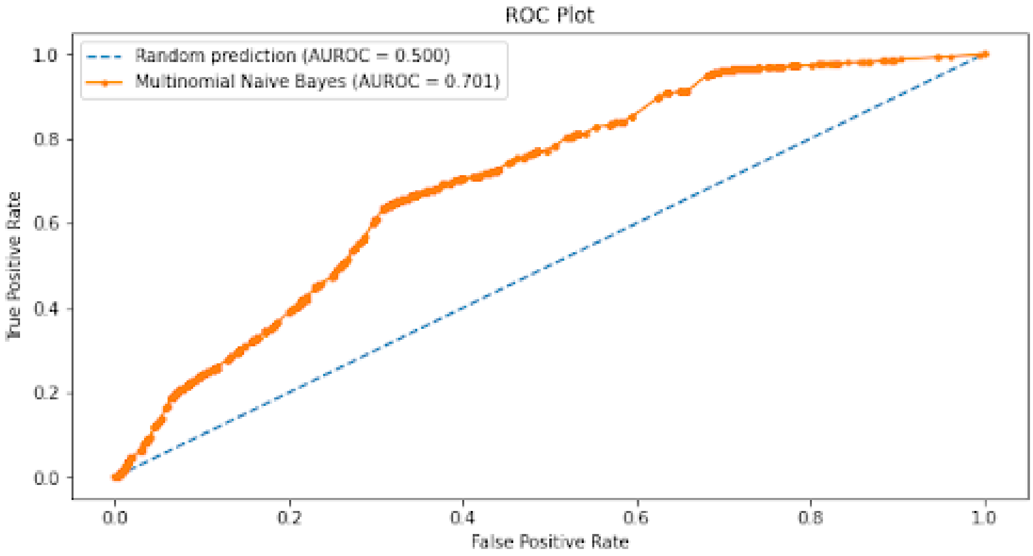


Figure 5.4: ROC curve generated using Multinomial Naive Bayes

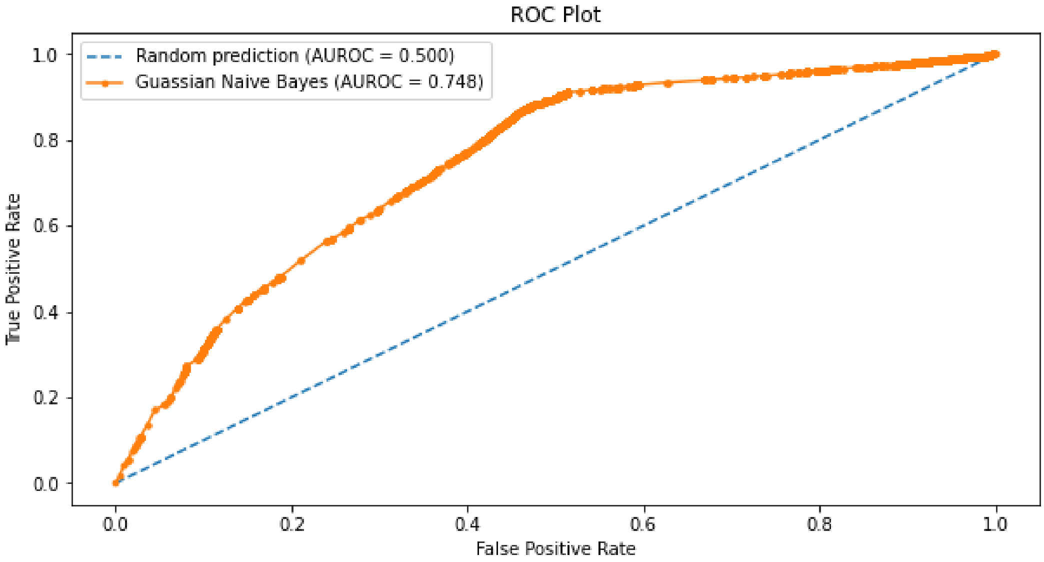


Figure 5.5: ROC curve generated using Gaussian Naive Bayes

Next, with the Naïve Bayes classifier, we get AUROC scores of 0.529, 0.701 and 0.748 for the Bernoulli, Multinomial and Gaussian variants respectively. The ROC curve generated using each of the Naïve Bayes variants is given above in Figures 5.3, 5.4 and 5.5 respectively.

Lastly, a comparison between the three algorithm's ROC curves can be observed below in Figure 5.6 (the Gaussian variant of the Naïve Bayes classifier is chosen for this comparison as it had the highest AUROC score).

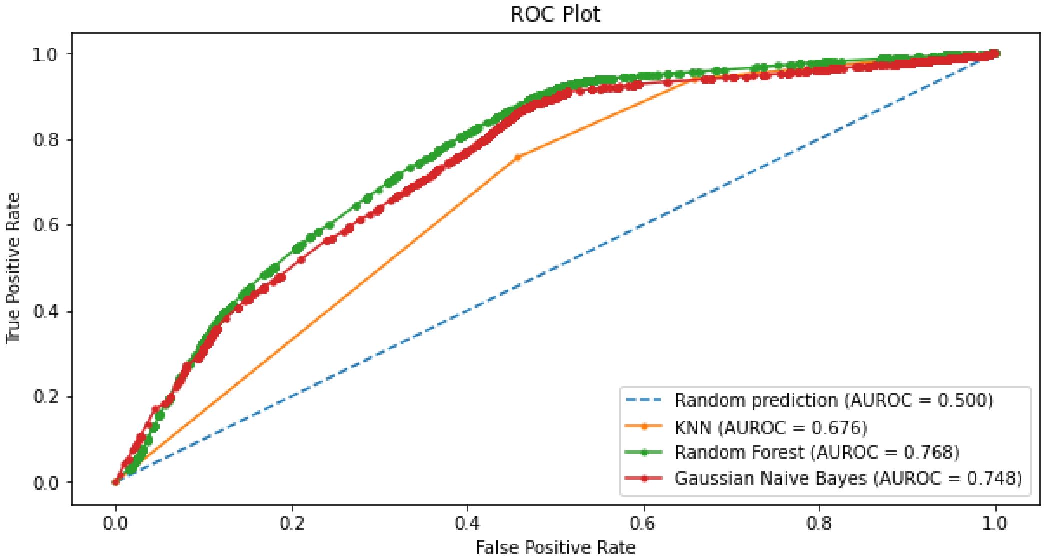


Figure 5.6: ROC curve comparison between RF, KNN and Gaussian Naive Bayes Classifiers

5.3 Time-split Analysis

As stated earlier, after the data was divided into the time splits of 3 month intervals, there were 5 variations of the train-test splits that were obtained.

For each of these cases, all the machine learning algorithms were once again employed and an accuracy score was obtained. The results are seen in the following sub-section.

5.3.1 Random Forest Time-split Analysis

The accuracy score obtained for each case while using Random Forest are described in Table 5.6 below.

Table 5.6: Accuracy score for each time-split train test case using Random Forest

Case Number	**Training Data**	**Testing Data**	**Accuracy Score**
1	From January 1, 2020, to March 31, 2020 (data of the first 3 months)	From April 1, 2020, to April 24, 2021 (all data after the first 3 months)	0.91
2	From January 1, 2020, to June 30, 2020 (data of the first 6 months)	From July 1, 2020, to April 24, 2021 (all data after the first 6 months)	0.93
3	From January 1, 2020, to September 30, 2020 (data of the first 9 months)	From October 1, 2020, to April 24, 2021 (all data after the first 9 months)	0.94
4	From January 1, 2020, to December 31, 2020 (data of the first 12 months)	From January 1, 2021, to April 24, 2021 (all data after the first 12 months)	0.94
5	From January 1, 2020, to March 31, 2021 (data of the first 15 months)	From April 1, 2021, to April 24, 2021 (all data after the first 15 months)	0.95

5.3.2 K-nearest Neighbor Time-split Analysis

When using K-nearest Neighbor, slightly different result's were obtained. They are described below in Table 5.7.

Table 5.7: Accuracy score for each time-split train test case using K-nearest Neighbor

Case Number	**Training Data**	**Testing Data**	**Accuracy Score**
1	From January 1, 2020, to March 31, 2020 (data of the first 3 months)	From April 1, 2020, to April 24, 2021 (all data after the first 3 months)	0.91
2	From January 1, 2020, to June 30, 2020 (data of the first 6 months)	From July 1, 2020, to April 24, 2021 (all data after the first 6 months)	0.93
3	From January 1, 2020, to September 30, 2020 (data of the first 9 months)	From October 1, 2020, to April 24, 2021 (all data after the first 9 months)	0.93
4	From January 1, 2020, to December 31, 2020 (data of the first 12 months)	From January 1, 2021, to April 24, 2021 (all data after the first 12 months)	0.94
5	From January 1, 2020, to March 31, 2021 (data of the first 15 months)	From April 1, 2021, to April 24, 2021 (all data after the first 15 months)	0.95

5.3.3 Naïve Bayes Time-split Analysis

Finally, the Naïve Bayes classifier for each of its variation gave its own results, as described below in Table 5.8.

Table 5.8: Accuracy score for each time-split train test case using Naïve Bayes

Case Number	Training Data	Testing Data	Accuracy Score (Bernoulli)	Accuracy Score (Multinomial)	Accuracy score (Gaussian)
1	From January 1, 2020, to March 31, 2020 (data of the first 3 months)	From April 1, 2020, to April 24, 2021 (till data after the first 3 months)	0.93	0.93	0.89
2	From January 1, 2020, to June 30, 2020 (data of the first 6 months)	From July 1, 2020, to April 24, 2021 (all data after the first 6 months)	0.93	0.93	0.91
3	From January 1, 2020, to September 30, 2020 (data of the first 9 months)	From October 1, 2020, to April 24, 2021 (alt data after the first 9 months)	0.93	0.94	0.91
4	From January 1, 2020, to December 31, 2020 (data of the first 12 months)	From January 1, 2021, to April 24, 2021 (all data after the first 12 months)	0.94	0.94	0.91
5	From January 1, 2020, to March 31, 2021 (data of the first 15 months)	From April 1, 2021, to April 24, 2021 (all data after the first 15 months)	0.95	0.95	0.91

A comparison of all three algorithms over the different cases of train and test data can be observed in the visual representation given below in Figure 5.7 (Multinomial variation of Naïve Bayes is used in place of other NB variations as it showed the better performance over the different cases.

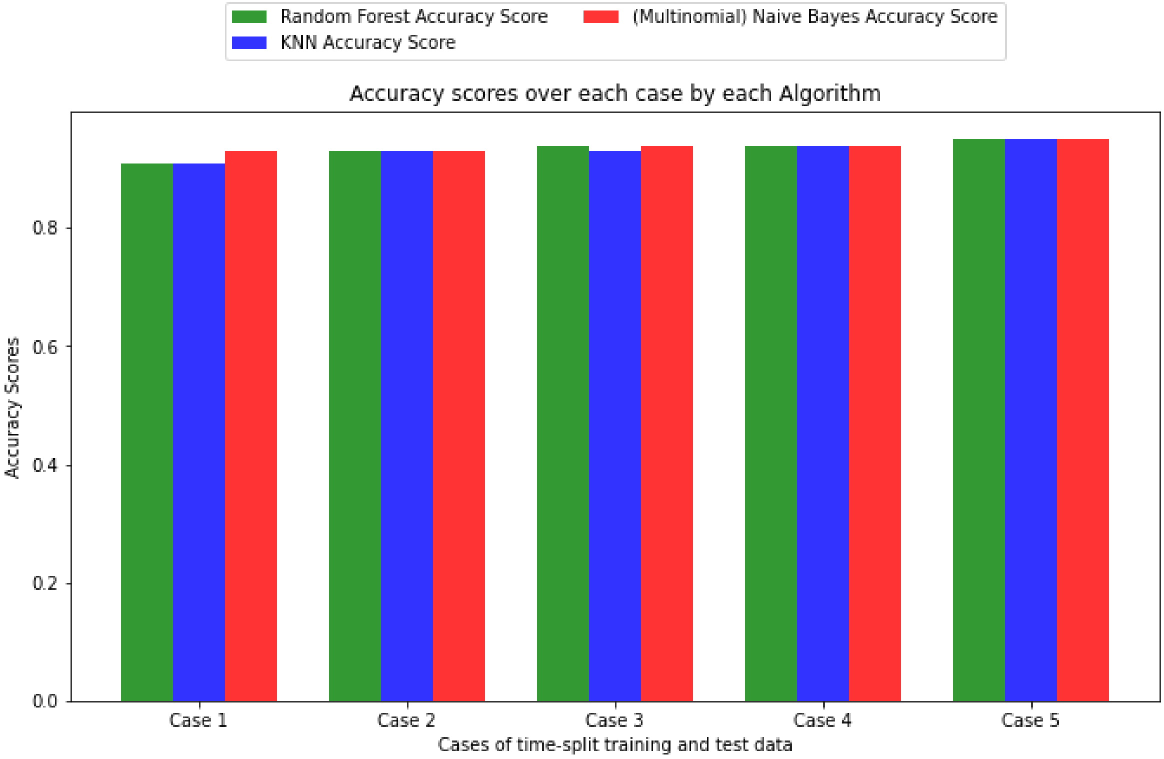


Figure 5.7: Algorithm performance over the different cases

5.4 Feature Data Analysis Results

The features selected to be analyzed over, were sex, age, intubed and pneumonia. These features were analyzed over the following time periods, as stated before:

- Time Period 1: January, 2020, to March, 2020
- Time Period 2: April, 2020, to June, 2020
- Time Period 3: July, 2020, to September, 2020
- Time Period 4: October, 2020, to December, 2020
- Time Period 5: January, 2021, to March, 2021

Additionally, three other features of obesity, inmsupr (immunosuppression) and and pregnancy were also recorded for each time period.

5.4.1 Total Patients meeting each Feature criteria

Over the entire dataset, the total number of patients that adhered to these features are as follows:

- Total number of Male Patients: 407,258
- Total number of Female Patients: 299,918

- Total number of Patients in Age Group 1 (0 - 18): 39636
- Total number of Patients in Age Group 2 (19 - 39): 100,856
- Total number of Patients in Age Group 3 (40 – 60): 260440
- Total number of Patients in Age Group 4 (61 and above): 306244
- Total number of Patients that needed to be incubated: 80778
- Total number of Patients with pneumonia: 420184

The results obtained for the data of patients meeting each feature criteria, in terms of each time period (TP), is given in Table 10 below. The "TP" column indicates the time period. The "Total" column indicates the total COVID-19 affected patients in that specific time period or TP. All subsequent columns indicate the number of patients that belong to that particular category of the column, within that TP.

Table 5.9: Total Patients that meet each feature criteria over each time period

TP	**Total**	**Male**	**Female**	**Age-group 1**	**Age-group 2**	**Age-group 3**	**Age-group 4**	**Intubated**	**Pneumonia**
1	9731	4929	4802	2359	2409	2515	2448	942	5407
2	146023	88301	57722	6167	23471	61487	54898	27341	89385
3	170863	98489	72374	9547	24388	63646	73282	21223	102148
4	174372	99405	74967	9467	24155	62388	78362	14871	100959
5	195590	110400	85190	11092	24705	67027	92766	14851	110782

5.4.2 Total Patients that meet Feature criteria and are also admitted to the ICU

The total number of patients that adhered to the aforementioned features and that also had to be admitted into the ICU are as follows:

- Total number of Male Patients that had to be admitted to the ICU: 33332
- Total number of Female Patients that had to be admitted to the ICU: 20544
- Total number of Patients in Age Group 1 (0 - 18) that had to be admitted to the ICU: 4492
- Total number of Patients in Age Group 2 (19 - 39) that had to be admitted to the ICU: 6612
- Total number of Patients in Age Group 3 (40 – 60) that had to be admitted to the ICU: 19930
- Total number of Patients tn Age Group 4 (61 and above) that had to be admitted to the ICU: 22782

- Total number of Patients what were intubated and that also had to be admitted to the ICU: 27078

- Total number of Patients with pneumonia that hat no be admitted to the ICU: 43840

The results obtained for the data of patients meeting each feature criteria and needing ICU support, in terms of each time period (TP), is given in Table 5.10 below. Like in Table 5.9, The "TP" column indicates the time period. The "Total" column indicates the total COVID-19 affected patients that needed ICU support in that specific time period or TP. All subsequent columns indicate the number of patients that needed ICU admittance and also adhered to that particular category (of the column), within that TP.

Table 5.10: Patients that meet each feature and need ICU support

TP	Total	Male	Female	Age-group 1	Age-group 2	Age-group 3	Age-group 4	Intubated	Pneumonia
1	644	396	248	95	123	224	202	368	557
2	13306	8562	4744	1148	1783	5424	4951	7667	11079
3	14506	8841	5665	1327	1804	5366	6009	7038	11754
4	12260	7571	4689	978	1438	4274	5570	5906	10086
5	11784	7164	4620	799	1246	4172	5567	5573	9406

As stated earlier, for more meaningful representation of the features that showed high correlation with the icu target variable, the percentage of patients that adhered to each feature and also needed ICU support out of all patients that associated with that feature was found for each time period. The resultant percentages, rounded to 1 decimal place, can be observed in Table 5.11 below.

Table 5.11: % of Patients that are associated with feature and who need ICU support out of total patients associated with the feature in each time period

TP	Intubated %	Pneumonia %	Obesity %	Immunosuppressed %	Pregnancy %
1	39.1	10.3	11.0	6.4	3.8
2	28.0	12.4	11.1	10.5	10.7
3	33.2	11.5	10.5	9.1	8.8
4	39.4	10.0	8.8	8.4	4.9
5	37.5	8.5	7.9	7.1	5.4

5.4.3 Feature Analysis Visualization

The total number of male and female patients over each time period is visualized in the bar chart below in Figure 5.8.

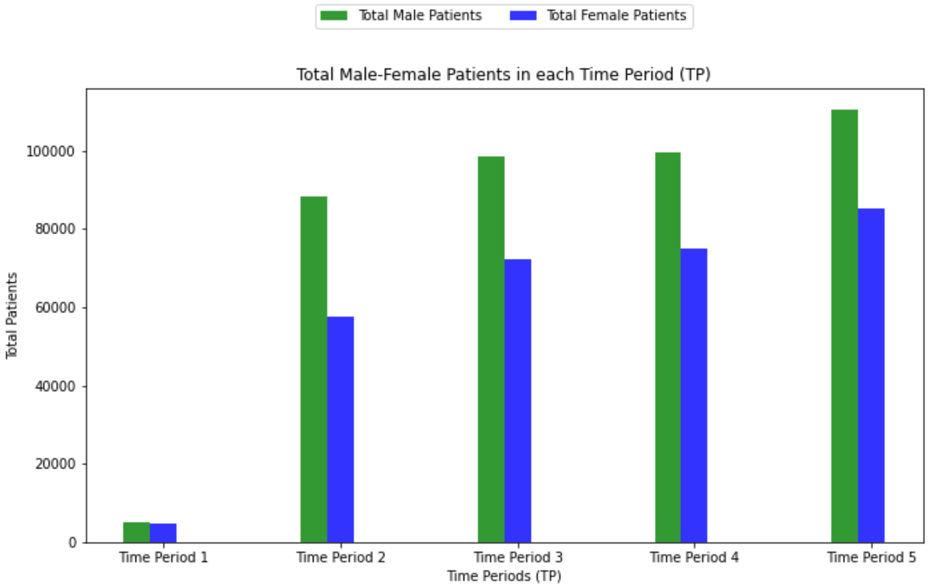


Figure 5.8: No. of Male and Female patients over each time period

Consecutively, a variation of that visualization, depicting the number of male and female patients that required ICU support over each time period can be observed in Figure 5.9 below.

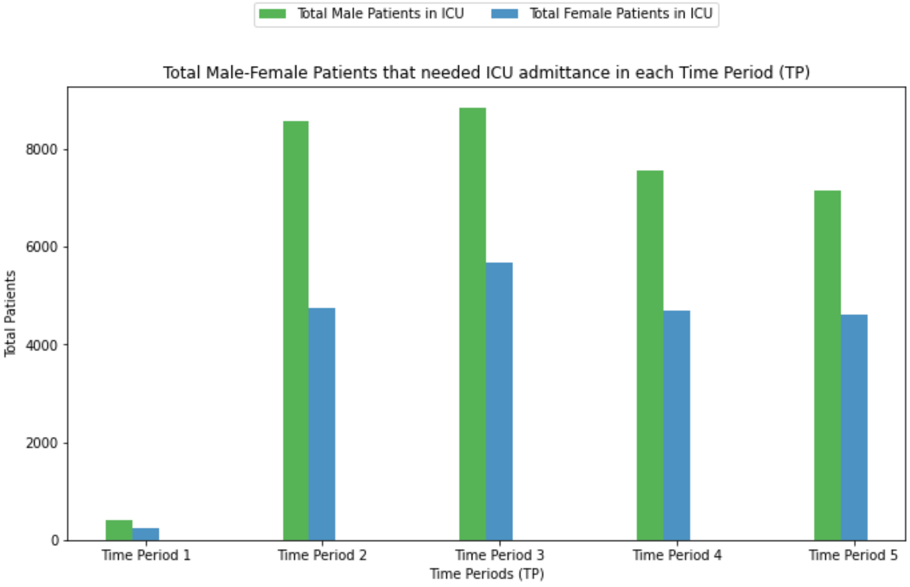


Figure 5.9: No. of Male and Female patients that required ICU admission over each time period

The bar chart depicting the total number of people belonging to each age group, spanning across each time period can then be observed in Figure 5.10 below.

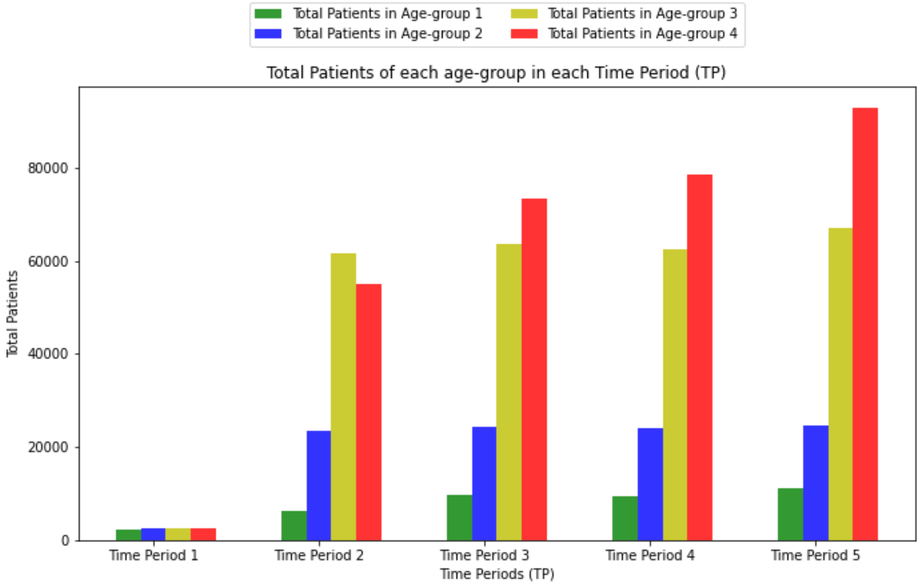


Figure 5.10: No. of patients belonging to each age group in each time period

Similarly, a different rendition of the age-group comparison, can be observed in the bar chart in Figure 5.11 below. It depicts the total number of patients of each age-group that had to be admitted into the ICU at each time period.

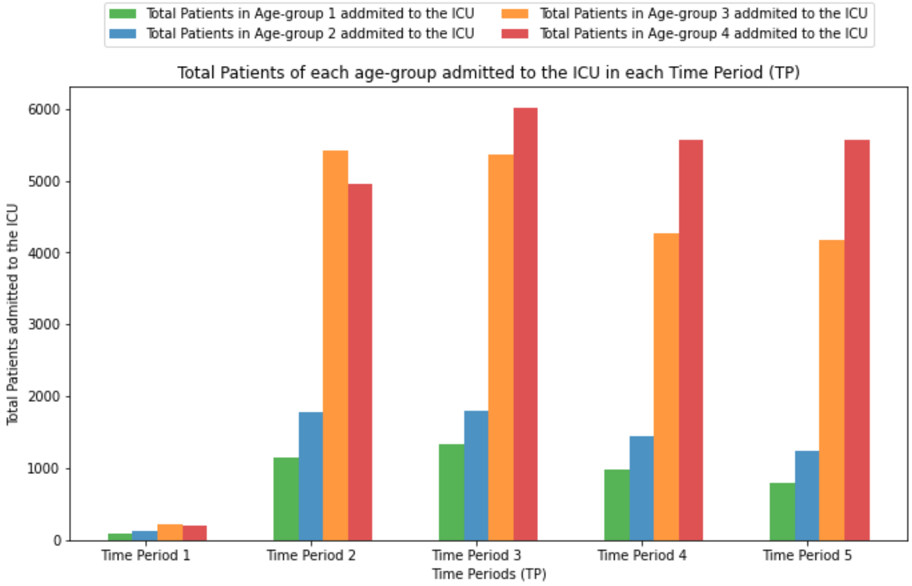


Figure 5.11: No. of patients of each age group that needed ICU support in each time period

Finally, a bar chart that visualizes the percentage of patients associated with the different features of high correlation and needing ICU support across all the time periods can be seen in Figure 5.12 below.

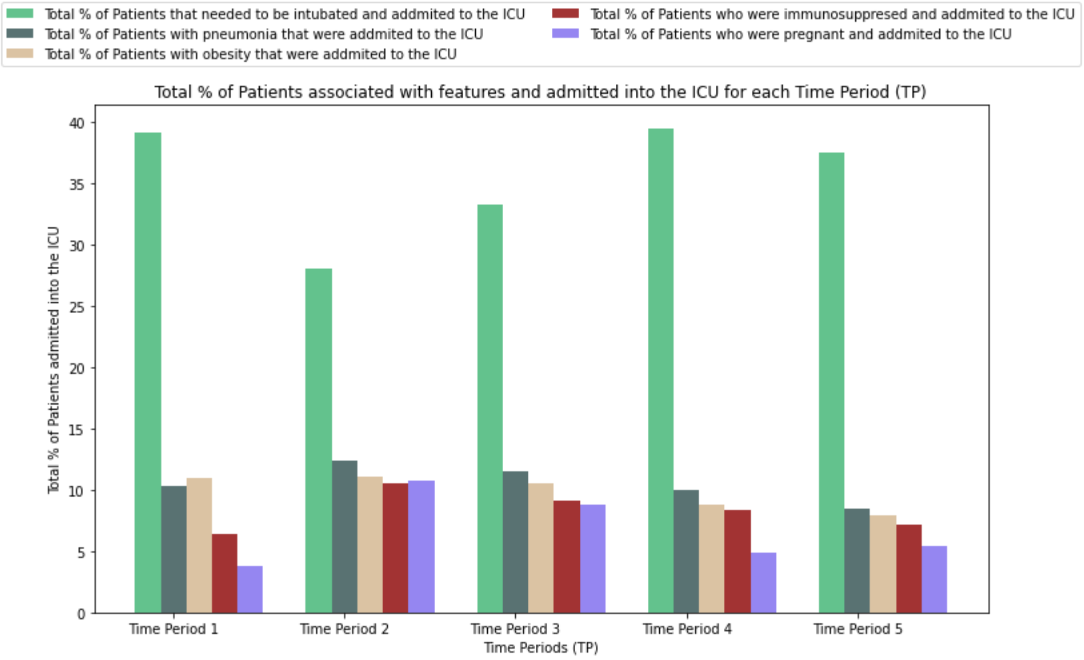


Figure 5.12: % of Patients associated with high correlated feature and ICU support out of total patients associated with the feature, in each time period

5.5 Implications

From the results that have been obtained – through the training and testing of machine learning algorithms, testing of time-split data and selected features over 3-month time periods – certain implications can be drawn about either the machine learning algorithm in use or the nature of the COVID-19 virus and the way it affects a patient based on certain characteristics.

5.5.1 Implications of Machine Learning Algorithm Results

The accuracy scores achieved by using each algorithm over the randomly split train-test data show that the Random Forest (RF) classifier, K-Nearest Neighbor (KNN) classifier, Multinomial variation of the Naïve Bayes (NB) classifier and the Bernoulli variation of the NB classifier all achieve results that are very similar to one another. Out of all of them, on pure accuracy score alone, the Multinomial variation of the NB classifier achieves the highest accuracy score of 0.92(6), with Random Forest right behind it at 0.92(5) and then the Bernoulli variation of NB classifier and the KNN classifier tying in at 0.92(4). The Gaussian variation of the NB classifier meanwhile had the lowest accuracy score out of all of them at 0.86(4).

On the metric of accuracy score alone, it would seem that the Multinomial NB classifier achieves the best results (through minimal differences), but the observation of the confusion matrices indicates a different outcome. The confusion matrix achieved by the Multinomial NB classifier strangely shows 0 true negative values and 0 false positive values. While the implication of the latter is that there are no false predictions of needing ICU support (which is a desired outcome), the implication of the former is that it failed to predict when someone did not actually need ICU support, which can be quite detrimental in real world applications. For that, it can be inferred that though on the basis of accuracy score, Multinomial Naïve Bayes did perform slightly better than the Random Forest classifier and KNN classifiers, the RF and KNN classifiers might still be a better choice when factoring in the confusion matrices as it has a score only slightly lower than the Multinomial NB, whilst still managing to correctly predict true negative values. The Bernoulli NB classifier also had similar scores, but it also showed a rather low accuracy in predicting the true negative values within its confusion matrix.

Another factor to consider when comparing the algorithms is the AUROC scores (and by extension the ROC curves). Out of all the classifiers, RF classifier shows the highest AUROC score at 0.768, followed by the Gaussian variant of the NB classifier at 0.748 and the Multinomial variant of the NB classifier at 0.701. The KNN classifier and the Bernoulli variant of the NB classifier are next with AUROC scores of 0.676 and 0.529 respectively. On the basis of AUROC scores alone, the RF classifier performs the best out of all the other algorithms. Thus, combining all the metrics of accuracy score, confusion matrices and AUROC scores (and ROC curves), it is a reasonable conclusion to draw that the Random Forest classifier does perform the better overall when compared to the KNN classifier or any variant of the Naïve Bayes classifier.

5.5.2 Implications of Time Split Analysis Results

The main reason for conducting the time-split analysis was to determine whether the model being proposed by this paper would still be relevant and accurate in making its predictions given the mutating nature of the COVID-19 virus. The train-test splits were varied on the basis of time in a manner such that the algorithm was trained on data from a set timeframe (or time split) and tested on data that came after that timeframe or time split. Doing this meant that any future variant of the COVID-19 strain (and the differences it can have on whether a person will end up needing ICU support or not) that did appear in the test set, would be unbeknownst to the model during its training phase. This meant that if the models were able to maintain a certain level of high accuracy or stability, then the possibility of future strains rendering the model to be less effective would be minimal. It is also worth consideration that the splits were ordered in a cascading manner, that is, for the first split, the training was done on the first 3 months and testing was done on the rest of the data. For the second split, the training was done for the first 6 months and then tested over the remaining data. Like this, the amount of data used for training was also varied in order to ascertain whether there was a significant effect on the model based on more thorough training data.

The results show that across all the classifiers of Random Forest, K-nearest Neighbor and Naïve Bayes (Bernoulli, Multinomial and Gaussian variants), there is a rather high accuracy score achieved throughout the time splits. All the classifiers show an accuracy score of at least 0.91 for all time cases of train-test data (except the Gaussian Naïve Bayes classifier, which shows an accuracy score of 0.89 for the first case). The best overall performance (in terms of accuracy score) was achieved by the Multinomial variation of the Naïve Bayes classifier. It is also worth noting that though relatively high accuracy was achieved with the algorithms even with three months of training data (as in the first case), the best results were unanimously obtained for all algorithms in the last case, where 15 months of training data were used to predict for just 1 month of testing data. Though having such a case may not be practical, the second best scores for the RF, KNN and NB (Bernoulli and Multinomial) were all at the 0.94, only slightly less than the best score of 0.95. Thus, there is not a very significant increase in accuracy by the expansion of the training split of data. The 0.94 accuracy score was achieved from training using data of the first 9 months (for RF and Multinomial NB) and the first 12 months (for KNN and Bernoulli NB) and then testing over the remaining data. With these results, it can be inferred that having a large enough data set that spans across a long enough timeframe can yield to having a model that is stable in making its predictions about patients needing extra medical attention, even with the mutating nature of the coronavirus.

5.5.3 Implications of Feature Analysis Results

The reason for conducting the feature based analysis was to get a deeper understanding of the nature of the COVID-19 virus in terms of how it affects people based on their characteristics (whether it was physical characteristics like age or sex, or pre-existing medical conditions). The entirety of the feature analysis was done on the basis of "time periods" or 3-month blocks that the entire dataset was divided into. The purpose for doing this was to investigate whether the spike in the number of cases of affected or the spike in the number of people needing ICU admission could be linked to any other feature or a collection of features or not.

When observing the different groups of sex different time periods, it can be noted that there is a greater number of male patients (affected by COVID-19) compared to female patients in each time period. As an extension, there is also a greater number of male patients needing ICU admission per time period compared to female patients. Then shifting over to the different age-groups across different time periods, it is observed that age groups 3 (40 – 60 years old) and 4 (61 years old and above) all have a significantly higher number of people affected by the virus and also needing ICU support compared to age groups 1 (0-18 years old) and 2 (19 – 39 years old).

A comparative analysis of features that showed high correlation to the target variable of needing ICU admission was also conducted. This was done in order to investigate just how likely it was for a person to be associated with a certain highly correlated feature and then need ICU admission (due to their condition worsening). The highest correlation was shown by the intubed feature. This was used to describe whether a patient needed intubation or not. Other highly correlated features were pneumonia, obesity, immunosuppression (whether the patient was immunosuppressed or not) and pregnancy. The percentage of patients who had to be intubated and then eventually taken to the ICU were significantly higher than any other patient adhering to any other feature and then taken to the ICU. It was significantly higher than even the percentage of patients with pneumonia (as a pre-existing medical condition, alongside COVID-19) needing ICU support even though pneumonia is the second highest correlated feature. For comparison, the highest percentage of patients with pneumonia and COVID-19 that required to be transferred to the ICU was 12.4% (in time period 2). Conversely, the lowest percentage of intubated patients that also transferred to the ICU was 28.0% (in time period 2). Intubation is a process that is required when a patient is unable to breathe on their own and a tube is inserted through the patient's throat and into their windpipe in order to allow air out of the patient's lungs. Since coronavirus is a respiratory disease, this high correlation with the need of intubation and the need of ICU support is expected, since intubation is also a sign of the person's condition worsening to the point where they would need ICU and ventilator support. Obesity was the next highest correlated feature and the percentage of obese patients that required ICU support was relatively close to the percentage of pneumonia patients that also required it. Percentage of immunosuppressed patients and pregnant patients that required ICU admission on the other hand was relatively lower.

When observing the spike in ICU admissions from time period 1 to during the time periods 2, 3, 4 and 5, there is a significantly higher number of patients being affected by the COVID-19 virus as this time can be considered its peak (at least in Mexico, where the dataset originates from). Though it can be simple to dismiss the high amount of ICU admission as simply due to the high amount of patients coming in overall, there are a few more nuances to it. For one, even though the overall spike is prevalent across both sexes and age groups, it is especially prominent for males and patients belonging to age groups 3 and 4. For reference, the number of female patients spiked from 4802 in time period 1 to 57,722 in time period 2. Meanwhile, the number of male patients rose from 4929 in time period 1 to 88,301 in time period 2. Though they both had similar numbers in time period 1, the spike was more prominent in time period 2 for male patients, meaning that more male patients were affected (or at least in need of hospitalization once affected). This resulted in greater number of male patients needing ICU admission (since there was greater number of male patients overall). Similarly, while age groups 1 and 2 spiked from 2359 and 2409 in time period 1 to 6167 and 23,471 respectively in time period 2, age groups 3 and 4 rose from 2515 and 2448 in time period 1 to 61,481 and 54,898 in time period 2. This spike varies once again from time period 2 to time period 3 where number of patients of age group 4 exceeds that of age group 3. This is also reflected in the ICU admission spike in the subsequent time periods as the greater number of patients from age groups 3 and 4 are the ones that end up needing ICU support.

Interestingly, even though the high correlation feature of pneumonia shows a much more prominent rise from time period 1 to 2 than that of the intubated feature, percentage wise, the number of intubated patients needing ICU support is significantly higher than the percentage of pneumonia patients needing ICU support.

Thus, even though the overall peak of the coronavirus causing greater amount of infections can be attributed to the rise in ICU admissions overall – it does not show the entirety of the situation. The way the virus is affecting people on the basis of sex, age or pre-existing medical condition (such as pneumonia, obesity and being immunosuppressed) or the greater number of people affected by the virus on the basis of those things, show a more detailed picture behind the spike in ICU admissions.

Chapter 6

Conclusion

The aim of this research was to create a model that would perform predictions on whether or not a person affected by the coronavirus would need extra medical assistance in the future or not due to their condition worsening. To reach that goal, this paper used the Random Forest, K-nearest Neighbors and Naïve Bayes machine learning algorithms over a dataset and tried to find the one that showed the best performance. On metrics of accuracy score, confusion matrices and AUROC scores, the Random Forest classifier showed a better performance overall compared to the other two. As an extension, data analysis was also performed on the dataset in order to pre-emptively address the issue of the model losing its relevance due to the mutating nature of the COVID-19 virus. Feature analysis was also performed in order to get greater insight into the nature of the virus and how it affects patients based on the patient's attributes (physical or medical). The research then, not only helped create a relatively high-performance prediction model that remained stable on variations of train-test data, it also provided valuable insight onto the nature of the COVID-19 virus and how it interacts with specific patients.

There were certain limitations in the model however. One was the lack of datasets available for both timeline 1 and 3, which limited the scope of what the research tried to accomplish – in creating a singular model that would serve as a predictor for COVID-19, a predictor for whether a person who was confirmed to have COVID-19 would need extra medical assistance in the future due to their condition worsening or not and a predictor for whether a person who had recovered from COVID-19 would develop post-recovery symptoms or not. For future additions and work on this research, the inclusion of those two timelines would result in a more holistic model. The results of the model in terms of both machine learning algorithm performance and also analysis, would also be more extensive with the addition of the other datasets. The performance of machine learning algorithms specifically would be an important factor as there would clearly be a singular algorithm that shows greater performance than the others over all datasets.

Another limitation was also that data used (for timeline 2)was from a singular geographic location in Mexico. Though the conditions being predicted on (such as whether a person requires ICU admission based on pre-existing medical conditions) are not necessarily related to the geographic location, using a similar dataset from different geographic location would further prove the robustness of the model and is something to be considered for future research. Expanding on this research on the basis of these limitations in the future could help produce a paper that can cover the bases that were not covered here.

Having a model in place that uses the best algorithm in place that can make the necessary predictions of whether a person has COVID-19 or not, whether a COVID-19 affected patient's condition will worsen or not and whether a recovered COVID-19 patient will have lasting health effects or not will help healthcare systems around the globe to facilitate medical resources efficiently and deliver treatment to people that need it most. It will also provide much needed information about the enigmatic ways in which the coronavirus correlates with various factors such as pre-existing health conditions, a person's age or sex and preliminary symptoms. Further analysis and research can develop on the basis of such findings and lead to much greater insight in to the nature of the virus itself. In short, having such a model will help with the management of medical resources in times such as this when it is as scarce, as well as help us better understand the COVID-19 virus as a whole.

Reference

[1] Osi Abdulhameed Ado et al. “A Classification Approach for Predicting COVID-19 Patient’s Survival Outcome with Machine Learning Techniques”. In: *medRxiv* (2020). DOI: 10.1101/2020.08.02.20129767. eprint: https://www.medrxiv.org/content/early/2020/08/10/2020.08.02.20129767.full.pdf. URL: https://www.medrxiv.org/content/early/2020/08/10/2020.08.02.20129767.

[2] Vaid Akhil et al. “Machine Learning to Predict Mortality and Critical Events in a Cohort of Patients With COVID-19 in New York City: Model Development and Validation”. In: *Journal of Medical Internet Research* 22 (Nov. 2020), e24018. DOI: 10.2196/24018.

[3] Sakifa Aktar et al. *Machine Learning and Meta-Analysis Approach to Identify Patient Comorbidities and Symptoms that Increased Risk of Mortality in COVID-19*. 2020. arXiv: 2008.12683 [q-bio.QM].

[4] Chansik An et al. “Machine learning prediction for mortality of patients diagnosed with COVID-19: a nationwide Korean cohort study”. In: *Scientific Reports* 10 (Oct. 2020), p. 18716. DOI: 10.1038/s41598-020-75767-2.

[5] Amit S. Gefen-Halevi S. Shilo N. Epstein A. Mor-Cohen R. Biber A. Rahav G. Levy I. Tirosh A. Assaf D Gutman Y Neuman Y Segal G. “Utilization of machine-learning models to accurately predict the risk for critical COVID-19.” In: *Intern Emerg Med.* (Nov. 2020), pp. 1435–1443. DOI: 10.1007/s11739-020-02475-0.Epub2020Aug18.PMID:doidoidoi=32812204;PMCID:PMC7433773.

[6] A. L. Booth, E. Abels, and P. McCaffrey. “Development of a prognostic model for mortality in COVID-19 infection using machine learning”. English (US). In: *Modern Pathology* 34.3 (Mar. 2021), pp. 522–531. ISSN: 0893-3952. DOI: 10.1038/s41379-020-00700-x.

[7] Awwalu Jamilu et al. “A MULTINOMIAL NAÏVE BAYES DECISION SUPPORT SYSTEM FOR COVID-19 DETECTION”. In: *FUDMA Journal of Sciences* 4 (June 2020), pp. 704–711. DOI: 10.33003/fjs-2020-0402-331.

[8] Das Ashis Kumar, Mishra Shiba, and Gopalan Saji Saraswathy. “Predicting CoVID-19 community mortality risk using machine learning and development of an online prognostic tool”. In: *medRxiv* (2020). DOI: 10.1101/2020.04.27.20081794. eprint: https://www.medrxiv.org/content/early/2020/05/13/2020.04.27.20081794.full.pdf. URL: https://www.medrxiv.org/content/early/2020/05/13/2020.04.27.20081794.

[9] Mohammadreza Nemati, Jamal Ansary, and Nazafarin Nemati. "Machine-Learning Approaches in COVID-19 Survival Analysis and Discharge-Time Likelihood Prediction Using Clinical Data". In: *Patterns* 1.5 (2020), p. 100074. ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2020.100074. URL: https://www.sciencedirect.com/science/article/pii/S2666389920300945.

[10] Subudhi Sonu et al. "Comparing Machine Learning Algorithms for Predicting ICU Admission and Mortality in COVID-19". In: *medRxiv* (2020). DOI: 10.1101/2020.11.20.20235598. eprint: https://www.medrxiv.org/content/early/2020/11/23/2020.11.20.20235598.full.pdf. URL: https://www.medrxiv.org/content/early/2020/11/23/2020.11.20.20235598.

[11] Souza Fernanda Sumika et al. "Predicting the disease outcome in COVID-19 positive patients through Machine Learning: a retrospective cohort study with Brazilian data". In: (June 2020). DOI: 10.1101/2020.06.26.20140764.

[12] Li Wei Tse et al. "Using Machine Learning of Clinical Data to Diagnose COVID-19". In: *medRxiv* (2020). DOI: 10.1101/2020.06.24.20138859. eprint: https://www.medrxiv.org/content/early/2020/06/24/2020.06.24.20138859.full.pdf. URL: https://www.medrxiv.org/content/early/2020/06/24/2020.06.24.20138859.

[13] Zoabi Yazeed and Shomron Noam. "COVID-19 diagnosis prediction by symptoms of tested individuals: a machine learning approach". In: *medRxiv* (2020). DOI: 10.1101/2020.05.07.20093948. eprint: https://www.medrxiv.org/content/early/2020/05/14/2020.05.07.20093948.full.pdf. URL: https://www.medrxiv.org/content/early/2020/05/14/2020.05.07.20093948.

[14] Cheng Fu-Yuan et al. "Using Machine Learning to Predict ICU Transfer in Hospitalized COVID-19 Patients". In: *Journal of Clinical Medicine* 9.6 (2020). ISSN: 2077-0383. DOI: 10.3390/jcm9061668. URL: https://www.mdpi.com/2077-0383/9/6/1668.