BUILDING PROGNOSTIC MODEL FOR COVID-19 OUTCOME USING MACHINE LEARNING TECHNIQUES

BY

EYASU BAHIRU SHIMELES

OFFICE OF GRADUATE STUDIES

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

August 31, 2021

Adama, Ethiopia

Building Prognostic Model for Covid-19 Outcome Using Machine Learning Techniques

By: Eyasu Bahiru Shimeles

Advisor: Teklu Urgessa (PHD.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfilment for the Degree of Masters of Science in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

August 31, 2021

Adama, Ethiop

ADVISORS APPROVAL SHEET

The advisor of the thesis entitled "Building Prognostic Model for Covid-19 Outcome Using Machine Learning Techniques" and developed by Eyasu Baheru**.** Hear by certifying that the recommendation and Suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis

Teklu Urgessa(Ph.D.) _____ _____

Advisor Signature Date

We, the undersigned, members of the board of Examiners of the thesis by Eyasu Baheru Shimeles**.** Have read and evaluated the thesis entitled "Building Prognostic Model for Covid-19 Outcome Using Machine Learning Techniques" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Masters of Science in Computer Science and Engineering.

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

Final approval and acceptance of the thesis are contingent upon submission of its Final copy to the office of postgraduate Studies (**OPGS**) through the Department Graduate Council (**DGC**) and School Graduate Committee (**SGC**).

_____ _____ _____

Department Head Signature Date

_____ _____ _____

School Dean Signature Date

Legesse Lemeche Obsu (PH.D.) _____ _____

Office of postgraduate studies. Dean Signature Date

Declaration

I hereby declare that this MSc Thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been properly acknowledged.

_____	_____	_____
Name Student	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that we have read the revised version of the thesis entitled "Building Prognostic Model for Covid-19 Outcome Using Machine Learning Techniques**"** prepared under my/our guidance by Eyasu Baheru Shimeles Submitted in partial Fulfillment of the requirement for the degree of master's degree of Computer Science and Engineering Therefore, I recommend submitting the revised of the thesis to the department following the applicable procedures.

_____	_____	_____
Advisor	Signature	Date

Acknowledgement

First and foremost, I'd like to thank God next to God I'd like to thanks my adviser Teklu Urgesa (PHD). Without his patience and understanding over the years, I would not have made it this far. His mentorship has helped me not only succeed academically but has also taught me much in my personal life. I have been very lucky to have him as my advisor.

I do not forget to thanks all my teacher from Elementary to University teacher, you have great contribution to my coming here. Special thanks to my aunt Mesnoshet Shimeles for her goodness and Positive advice.

My all classmates, friends, and colleagues in and outside the University are all acknowledged for their friendship and sharing life together during my stay at Adama Science and Technology University; particularly. Last but not least I offer my special thanks for those involve directly or indirectly in this study and their names are not listed

Table of Contents

List of Tables

List of Equations

List of Figure

List of Abbreviations

AI	Artificial Intelligence
AUC	Area Under Curve
AUPRC	Area Under Precision Recall Curve
AUROC	Area Under Receiver Operating Characteristic Curve
ECG	Electrocardiogram
EHR	Electronic Health Record
FN	False Negative
FP	False Positive
KNN	K-Nearest Neighbor
ICU	Intensive Care Unit
LR	Logistic Regression
ML	Machine Learning
MLP	Multiline Perceptron
RF	Random Forest
ROC	Receiver Operating Characteristic Curve
SVM	Sector Vector Machine
SHAP	Shapley Additive exPlanations
SPHMM	St.Pauls Millennium Medical College
TN	True Negative
TP	True Positive

Abstract

The global covid-19 pandemic puts great pressure on medical resources worldwide and leads healthcare professionals to question which individuals are in imminent need of care. With appropriate data of each patient, hospitals can heuristically predict whether or not a patient requires death or survive from the pandemic. We adopted a machine learning model to prognostic of individuals who tested positive given the patient's underlying health conditions, age, sex, and other factors. As the allocation of resources toward a vulnerable patient could mean the difference between life and death, a prediction model serves as a valuable tool to healthcare workers in prioritizing resources and hospital space. In this work, we use the patient demographics, laboratory data and clinical reports as the predictors. The used models are the random forest, sector vector machine, K- Nearest Neighbor, logistic regression and multiline perceptron. In our experiment, we use Confusion matrix, precision, accuracy, and f1-score for performance metrics. RF score better accuracy from the selected machine learning models, the result of RF shows (accuracy =97.87, precision = 0.8, F1-score = 0.44, Recall = 0.51). Results indicate that the RF model outperforms form other machine learning models.

Keywords:

Machine Learning, LR, SVM, KNN, RF, MLP, COVID-19, prognostic model

CHAPTER ONE

1. INTRODUCTION

1.1. Background of Covid-19

Covid-19 is a new disease, caused by a type of virus named severe acute respiratory syndrome coronavirus [1]. Coronaviruses are a family of viruses that can cause problems with the respiratory system. When this new virus infects someone, the person may or may not have any symptoms. If a person does have symptoms, those symptoms severity can range from mild to severe [2]. The most common symptoms are fever, dry cough, tiredness, sore throat, and shortness of breath, this list not all-inclusive, those symptoms usually appear 2–14 days after the person is infected with the virus [2]. According to CDC report majority of those who die in a COVID-19 have pre-existing conditions, including cancer, hypertension, diabetes, cardiovascular disease, smoking, and obesity [3].

The SARS has spread across all continents since and caused a public health crisis. The first report of SARS-CoV-2 was in November 2019, in Wuhan, China. On August 8 2021, over 204,971,119 million people had confirmed coronavirus disease worldwide and at least 4,330,821 people had died from the disease [4].

In Ethiopia, 587 new infections are reported on average each day. That's 28% of the peak the highest daily average reported on April 6. There have been 285,413 infections and 4,440 coronavirus-related deaths reported in the country since the pandemic began. Ethiopia has administered at least 2,291,339 doses of COVID vaccines so far [5]. Assuming every person needs 2 doses, that's enough to have vaccinated about 1% of the country's population. Ethiopia averaged about 7,862 doses administered each day. At that rate, it will take a further 2,852 days to administer enough doses for another 10% of the population [6].

Although a shortage of testing kits in epidemic areas increase the screening burden, and many infected people are thereby not isolated immediately, this accelerates the spread of COVID-19. On the other hand, due to the lack of medical resources, many infected patients cannot receive immediate treatment [7].

To mitigate the burden on the healthcare system, while also providing the best possible prognostic model is mandatory. The ongoing public health emergency necessitates the discovery of reliable prognostic models to guide clinical decision making and treatment plans tailored to the patient characteristics. These prognostic models could also improve the design

and analysis of future clinical trials and suggest novel insights into the disease [8]. In a prognostic model, multiple predictors are combined to estimate the probability of a particular outcome or event (for example, mortality, disease recurrence, complication, or therapy response) occurring in a certain period in the future. This period may range from hours, weeks, months or years [9] [10].

Traditionally, standard statistical methods and doctor's intuition, knowledge and experience had been used for prognosis. This practice often leads to unwanted biases, errors and high expenses, and negatively affects the quality of service provided to patients [10]. With the increasing availability of electronic health data, more robust and advanced computational approaches such as machine learning have become more practical to apply and explore in disease prognostic area.

Machine learning has been applied to many areas in health care, including image diagnosis, digital pathology, prediction of hospital admission, drug design, classification of cancer and doctor assistance, etc. Machine learning enables us to build prognostic models that help doctors to predict the outcome of a disease, choose the best possible treatment for each patient and allow for effective allocation of health resources [11]. In the literature, most of the related studies utilized one or more machine learning algorithms for a particular disease prognostic. For this reason, the performance comparison of different supervised machine learning algorithms for disease prognostic is the primary focus of this study.

After a comparison of different supervised ML model, focused on the implementation of ML method to support medical decisions support. Prognostic models combine multiple prognostic factors to estimate the risk of future outcomes in individuals with a particular disease or health condition. A useful model provides accurate predictions to support decision making by individuals and caregivers. Using established data, historical, clinical, and investigational variables are identified systematically and combined in a model to estimate the probability of an outcome. [11]

In Ethiopia, machine learning research on health care is almost nonexistent but it is believed that if machine learning is applied in this area, it might be critically important in revealing a decision-support system. Therefore; the researcher is motivated to see the potential applicability of prognostic machine learning models on COVID-19 data.

1.2. The Motivation of the Study

We are living through unprecedented times. The impact of the novel coronavirus has reverberated through every corner of the globe taking lives, destroying livelihoods, and changing everything about how we interact with each other. The sudden increase in COVID-19 cases is putting high pressure on healthcare services worldwide [7].

At this stage, fast, accurate and early clinical assessment of the disease severity is vital. When we see people infected with the covid-19 virus will be experienced mild to moderate respiratory illness and recover without requiring special treatment [2]. Older people and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illnesses [3].

However, many ongoing clinical trials are evaluating potential treatments. Currently, there are no validated prognostic models or scoring systems applicable specifically to patients with SARS-CoV-2, despite attempts to set general predictors of mortality. Emerging clinical risk scores have been limited by small sample sizes, this model is used to create an online calculation tool designed for patient triage at admission to identify patients at risk of severe illness, ensuring that patients at greatest risk of severe illness receive appropriate care as early as possible and allow for effective allocation of health resources.

1.3. Statement of the Problem

The pandemic of COVID-19 posed a challenge to global healthcare. According to WHO reports more than 4,350,690 patients have died from covid-19 [4]. It has been reported that the majority of those who die on covid-19 have pre-existing conditions including cancer, hypertension, diabetes and cardiovascular disease. [3].

Researches on the effects of COVID-19 are increasing, most related investigations have a little example size, and limited information can’t give a full image of the risk. Secondly, the studies many of them are from China, so the generalizability of findings to other countries is not clear. The Chinese may differ from other populations in terms of their health-seeking behaviour, symptom reporting, the prevalence of different comorbidities, as well as their living lifestyle [12] [13] [14]. However, there are significant differences between China and other countries.

Machine learning algorithms can analyze a large number of parameters within a short period to identify the predictors of disease outcomes. Electronic health records (EHR) are one of the main sources of data in the field. They typically include multiple types of clinical data (i.e., demographics, laboratory data and comorbidities) and aims to contain complete records of a

patient's medical history. When working with EHR, we must face problems related to uneven data quality, the presence of both structured and unstructured data and extreme variability problems.

However, due to the shortcomings in current prognostic methods has the accuracy of the predicted probability seems questionable, generalizability problems and lack of quality of data. Hence, the problem of the statement is how to improve those problems based on our prognostic model.

Research questions

To achieve the objectives of this research, the answers to the following questions.

- Which machine learning model can be used to prognostic COVID-19 with better accuracy?
- What are the features that will influence the prognostic result of COVID-19?

1.4. The Objective of the Study

1.4.1. General Objective

The general objective of this study is to Building Prognostic Model for Covid-19 Outcome Using

Machine Learning Techniques.

1.4.2. Specific Objectives

The specific objectives of the research are identified as follows:

- Gathering demographical, comorbidity, symptoms and laboratory data's
- To review different kinds of literature that support the study in the area of applying machine learning approaches for COVID-19 spread prediction.
- Develop a predictive model based on early triage data result
- Providing clinical decision support knowledge
- Help Health service planning and effective allocation of health resources based on extracted knowledge

1.5. Scope and Limitation of the Study

1.5.1. The Scope of the Study

The study focuses on designing and implementing a prognosis model that can automatically predict, or estimating the probability or risk of future conditions

1.5.2. The Limitation of the Study

Our study has several limitations: First, the sample size was relatively small, and may not fully reflect the characteristics of the disease. Therefore, a large sample size could give a more comprehensive understanding of Covid-19. Second, the study findings might have been biased by reporting only confirmed cases in two hospital centers. Third, we statistically analyzed the laboratory findings based on a comparison of means, median, and proportions between different age groups that were not subdivided into groups of patients with individual comorbid conditions. Finally, the study assessed the epidemiological, laboratory, and clinical characteristics of COVID-19 on the admission of the patients; more detailed information from other laboratory tests and clinical outcomes were unavailable at the time of analysis.

1.6. Application of the Study

The main use of this study is the physician who takes action and makes decisions on COVID-19 patients. Also, it helps to reduce the mortality rate and helps to utilize resources properly in the right place depending on the outcome of this study. Additionally, researchers can replicate the proposed research for other infections in Ethiopia and can serve as a baseline for related work on this issue or use the dataset to improve the research.

1.7. Organization of the Thesis

This research paper is organized into seven chapters in the following ways:

Chapter One: discussed above and includes the introduction of the study, the motivation, and the statement of problems, the research questions that would be answered in the proposed solutions, the scope and limitation, the objective of the study, and the application of the study.

Chapter Two: presents the literature review, prognostic model, prognostic factors, clinical decision support system, machine learning algorithms, and related works on the COVID-19 prognostic model,

Chapter Three: discusses the research methodology for this research work, methods used to build the dataset, preprocessing steps (data cleaning and transformation), and machine learning approach, finally, it presents the evaluation methods.

Chapter Four: discusses the proposed framework used to build a prognostic model for COVID-19 outcomes, Dataset Description, Descriptive Analysis, Preprocessing, Dimensional Reduction, Models, Models Evaluation and Testing.

Chapter Five: Discusses the Implementation and Experimentation of the proposed solution, Data Environment Setup, Descriptive Analysis, Data Preprocessing, Feature selection, Model Building Evaluation

Chapter Six: Discusses the Results of the proposed machine learning approaches and also discusses the major results obtained by comparing all models based on the performance metrics.

Chapter Seven: Conclusion, Recommendation, Future Work.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

This chapter contains the background theory and related words required for the implementation of the thesis work. Details of the chapter focus on prognostic corona virus clinical outcomes, based on early triage data. The review covers concepts of prognostic clinical factors, prognostic models, machine learning, machine learning algorithms and related work.

2.2. Clinical Prognosis

Prognosis is the prediction of the course of disease following its onset. It refers to the possible outcomes of disease and the frequency with which these outcomes can be expected to occur [15]. Here, the event is a general term that captures a variety of things that can happen to an individual. Events can include outcomes such as death and other adverse events like a heart attack or a stroke, which might be risks for patients who have a specific medical condition or for the general population.

Making a prognosis is a clinically useful task for a variety of reasons:

- First, the prognosis is useful for informing patients of their risk of developing an illness.
- Second, the prognosis is also useful for informing patients how long they can expect to survive with a certain illness. An example of this is cancer staging, which gives an estimate of the survival time for patients with that particular cancer.

Sometimes the characteristics of a particular patient can be used to more accurately predict that patient's eventual outcome. These characteristics are called prognostic factors.

2.3. Prognosis factors

Prognosis factors can be used to predict an outcome. Prognostic factors need not necessarily cause the outcomes, just be associated with them strongly enough to predict their development [16].

Table 2-1 Prognostic factors

No.	Prognostic factor	Types
1	Demographic	- Age - Gender - Obesity

		▪ Smoking history
2	Comorbidity	▪ Hypertension ▪ Cardiovascular disease ▪ Cerebrovascular disease ▪ Peripheral artery disease ▪ Dementia ▪ Diabetes ▪ Chronic respiratory disease (e.g., COPD, obstructive sleep apnea) ▪ Active malignancy ▪ Immunosuppression ▪ Chronic kidney or liver disease ▪ Rheumatologic disease ▪ Bacterial or fungal coinfection
3	Symptoms	▪ Myalgia ▪ Pharyngalgia ▪ Sputum production ▪ Chills ▪ Nausea ▪ Dyspnea ▪ Chest tightness ▪ Dizziness ▪ Headache ▪ Hemoptysis ▪ Tachypnea ▪ Hypoxemia ▪ Respiratory failure ▪ Hypotension ▪ Tachycardia
4	Laboratory and other investigation	▪ Lymphopenia ▪ Leukocytosis ▪ Neutrophilia

		• Thrombocytopenia • Hypoalbuminemia • Liver or kidney impairment • Elevated inflammatory markers (C-reactive protein, procalcitonin, ferritin, erythrocyte sedimentation rate) • Elevated lactate dehydrogenase • Elevated creatine kinase • Elevated cardiac markers • Elevated D-dimer • Elevated interleukin-6 • $PaO_2/FiO_2 \leq 200$ mmHg
5	Complications	• Shock • Acute infection or sepsis • Acute kidney, liver, or cardiac injury • Acute respiratory distress syndrome • Venous thromboembolism • Arrhythmias • Heart failure

2.4. Clinical Prognostic Models

In medicine, numerous decisions are made by care providers, often in shared decision making, on the basis of an estimated probability that a specific event will occur in the future or prognostic setting in an individual. In the prognostic context, predictions can be used for planning lifestyle or satisfying decisions on the basis of the risk for developing a particular outcome or state of health within a specific period [9] [10]. Such estimates of risk can also be used to risk-stratify participants in beneficial intervention trials [17].In prognostic setting, probability estimates are commonly based on combining information from multiple predictors observed or measured from an individual [9] [10]. Information from a single predictor is often insufficient to provide reliable estimates of prognostic probabilities or risks [18].In virtually all medical domains, prognostic risk prediction models are being developed, validated, updated, and implemented with the aim to assist doctors and individuals in estimating probabilities and potentially influence their decision making. A multivariable prediction model is a mathematical equation that relates multiple predictors for a particular individual to the probability of or risk for future

occurrence of a particular outcome [19].Predictors are also referred to as covariates, risk indicators, prognostic factors, determinants, test results, or more statistically independent variables. They may range from demographic characteristics (for example, age and sex), medical history taking, and physical examination results to results from imaging, blood and urine measurements.

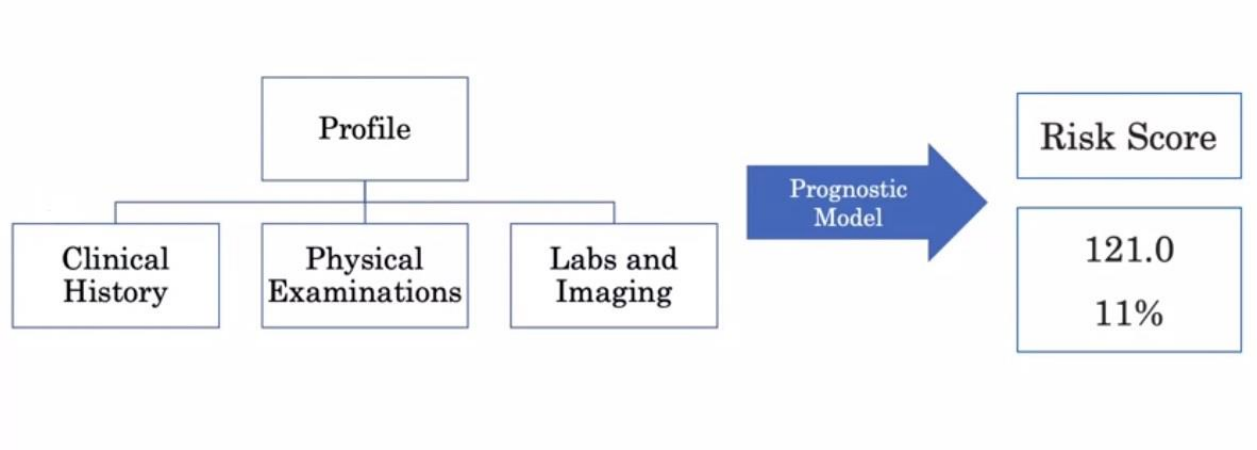

Figure 2-1 Prognostic risk score system diagram

From the above block diagram, we can see that set of features are patient profile includes clinical history, physical examinations and labs and imaging and the target would be risk score computed from the features.

2.5. Clinical Decision Support System

Medical prognosis conducted by physicians today tend to be highly subjective and conform based on their intuition, experiences, judgement, emotions, and knowledge. Improved by the fact that medical history, clinical data and symptoms often follow a linear relationship, and the expected outcome at the individual level does not always acceptable to the rules of epidemiology; the healthcare industry must adopt a more objective approach (Chattopadhyay, 2013). One method that has been proposed is the use of computational machine learning techniques that allow the extraction of interesting, meaningful and predictive information from clinical data.

This approach has the potential to:

- Eradicate some degree of physician's subjectivity
- Allow the epidemiology to work more precisely at the patient level
- Enable a more comprehensive set of data to be analyzed simultaneously
- Ensure a more objective output to be generated

However, it is noteworthy that the final clinical decision should be made by the physicians as humans are more flexible and capable of identifying outlying details that CDSS is unable to account for due to the lack of certain information. Hence, CDSS should serve as guidelines aiming to leverage the overall standard of healthcare and should not be used as a replacement for physicians. An ideal scenario is to capitalize on the highly accurate prediction that machine learning-based CDSS can offer while allowing physicians to have full flexibility and responsibility in making a good clinical judgement [20]. Moreover, it has been realized that CDSS do offer significant advantages like improve patient safety, quality of care, and efficiency in healthcare delivery. When deployed appropriately [21]. Hence, a vital task is to accurately identify those aspects of clinical practice that are best suited for their introduction. These promises have anticipated the current union of interests on the employment of artificial intelligence (AI) and statistical modelling as computational reasoning tools to support clinical decisions. These approaches have the distinct advantages of performing non-linear inference, exploratory data analysis, tolerating noise, avoiding the difficulties of acquiring expert knowledge and the ability to accommodate and model new manifestations of the disease. Given these promising benefits, many CDSS has been developed in recent years using ML methods. This approach empowers users to automatically discover the underlying medical knowledge from large medical databases that could be stored in different sources through the process of learning from experiences. This process of learning allows the performance of certain tasks to improve over time with experience; here, experience refers to the data that is used for training the ML inference model. In other words, the algorithm will search through the possible hypotheses within the boundaries of the selected mathematical or computational model to identify the one that best suits the observed data and any prior knowledge possessed by the learning algorithm. The nature of the data, in this case, can be described by nominal or numerical information called attributes (e.g. gender, age, family history, etc.) and time-series information (e.g. electrocardiogram, blood pressure, etc). If an algorithm is allowed to learn from more data, it will gain more experience. Similarly, if high-quality data is presented to a classification algorithm, a good experience will be gained.

2.6. Machine Learning

2.6.1. Machine Learning Technique

Machine learning (ML), a term coined by Samuel in the 1950s, is concerned with the creative design and development of learning procedures capable of empowering computers with the

ability to autonomously learn to solve a problem without explicitly being programmed [22]. The aim of machine learning research in healthcare is not to replace human doctors or nurses, but rather to supplement and provide support where humans struggle. By doing precisely what humans can't, namely processing huge amounts of data quickly, machine learning methods can both improve the quality and consistency of care on a large scale [23] [24].

ML is broadly classified into four types:

Supervised Machine Learning: -the algorithm is allowed to learn from a data set with pre-defined labels. Classification and regression are the two main types of supervised learning.

Unsupervised Machine Learning: -These algorithms attempt to learn from unlabeled data sets. The algorithms work by processing the unlabeled data set to extract features and identify patterns. Examples of unsupervised ML algorithms include clustering and dimensionality reduction of large and high-dimensional data sets.

Reinforcement Machine Learning:-In reinforcement learning, the algorithm learns through trial and error. Thus, a reward and punishment mechanism is employed in the training phase.

2.6.2. Machine Learning Algorithms

During our research, we have investigated regression and classification algorithms through which we have performed a better prognostic model.

K-nearest-neighbor

K-nearest neighbor (KNN) is a model that classifies input variables by doing a majority vote of its neighbours. The case is assigned to the majority class measured by a distance function (similarity measure). If there is only one neighbour, then the object is assigned to the neighbour's class that is closest to the point. The number of neighbours that are considered for the majority vote in the algorithm is a number K. K is a hyperparameter that must be set in the beginning and is usually chosen by first inspecting the data. A large K usually is more accurate and reduces the variance that exists in the data. There is no perfect way of finding optimal K, but can be retrospectively determined using hyperparameter tuning algorithms like cross-validation with the use of a validation dataset.

Support vector machine

A Support Vector Machine (SVM) is a model that can be used for both classification and regression tasks. The algorithm marks each data point into one of two classes. To find out which

data point belongs to each class the algorithm finds a hyperplane that differentiates the data points of both classes by the largest possible margin [25].

Logistic Regression

Logistic regression is named after the logistic function, often referred to as the sigmoid function in machine learning. This function is used at the core of the algorithm to calculate a probability value between 0 and 1 that can be mapped to two or more discrete (only specific values or categories are allowed) classes. If the probability value of a data point towards a certain class exceeds a set threshold it is categorized as that particular class [26].

Multilayer Perceptron (MLP): is a known and most used neural network which can be used for regression problems. This model has multiple layers consisting of neurons. Learning in this model is achieved in a supervised manner. The power of MLP comes from the non-linear activation function. There are activation functions for updating the weight in each layer. The three layers in the network are the input layer, hidden layers, and the output layer. The choice of activation function for the output layer depends on the nature of the problem to be solved. For the hidden layers of neurons, sigmoid functions are preferred, because they have the advantage of both non-linearity and differentially. For output neurons, the activation function adapted to the distribution of the output data is recommended [27]

Random forest (RF): is a DT ensemble method for classification tasks output of random forest is the selected by most tress. For regression tasks, the mean or average prediction of the individual trees is returned. That creates multiple trees through a re-sampling process called bagging (bootstrap aggregation). Numerous DTs are constructed by re-sampling using bootstrapping with replacement. Each node of the tree is split using a subset of the attributes that are selected randomly for each tree. Class membership for a new example is identified as the most commonly predicted class from the DTs by a simple unweighted majority vote [28] [29].

Table 2-2 Different ML algorithms pros vs cons

Machine Learning	**Basic ides**	**Pros**	**Cons**
SVM	Hyperplane optimization	Effective in high dimensional spaces.	If the number of features is much greater than the

		Still effective in cases where number of dimensions is greater than the number of samples. Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.	number of samples, avoid over-fitting in choosing kernel function and the regularization term is crucial. SVMs do not directly provide probability estimates
KNN	Based on a distance metric to measure the distance between data points.	Simplicity:- Very easy to implement. Non parametric, Very sensitive, Versatility	KNN slow algorithm Curse of Dimensionality KNN need homogeneous feature Outlier Sensitivity Missing Value treatment
LR	Predicts the probability that a given data point belongs to certain class	Probability prediction, Thrives with Little Training, Efficient Computation, Reputation is king, Unlikely to over fit, Model Flexibility	Overfitting Possibility, Regularization, Limited Use Case, Linear Decision Boundary, High Data Maintenance, Can't Handle Missing Data
MLP	Neural Network Method	Can be applied to complex non-linear problems	It is not known extent each independent variable is affected by

		Works well with large input data. Provides quick predictions after training The same accuracy ratio can be achieved even with smaller data.	the dependent variable. Computation are difficult and time consuming The proper functioning of the model depends on the quality of the training
RF	Decision Tree ensemble method	Effective for highly complex problems, best for high-dimensional data sets, can handle missing data and imbalanced data sets, Excellent Predictive Powers, Optimization Options	Overfitting Risk, Limited with Regression, Parameter Complexity, Biased towards variables with more levels, TradeMark situation

2.7. Related Works

The effort has been put into the development of the prognostic model to predict the risk of critically ill patients. Health-based systems are very critical, hence it is important to predict them accurately. Several health care systems use computer-aided clinical predictive models like the risk of heart failure [30], mortality in pneumonia [31], mortality risk in critical care [32]. Importantly, machine learning predictions will need to be transparent to ensure medical professionals embrace this new technology. Such a comprehensible deconvolution allows medical professionals to combine the temporal predictions with their existing beliefs to facilitate decision making. Such temporal rankings of features might assist medical professionals in deciding on the timing of interventions during admissions. The study shows that it is possible to make dynamic and easily interpretable models that predict mortality in critically ill patients. Such models can deliver new insight into complex interactions, non-linearity, and the importance of trends in the explanatory variables. This kind of model can be used as a decision support tool, the results need to be confirmed in a prospective clinical trial.

Booth et al [33]. Developed a machine learning model to predict mortality in COVID-19-positive patients using clinical and laboratory data's. In this study the data set was collected from 398 patients (355 survivors and 43 non-survivors from COVID-19) to predict death up to 48 hours in advance. The author ML techniques, LR and SVM to build the prediction model. From the 26 parameters that were initially collected, the top five highest-weighted laboratory values were then selected CRP, BUN, serum calcium, serum albumin, and lactic acid. The paper shows SVM model achieved 91% sensitivity and 91% specificity (AUC 0.93) for predicting patient death.

Shaikh Fs-et al [34]. Developed a Cox proportional hazard model to predict top three comorbidities, which have a high contribution for severity. The data set was collected from Prince Mohammed Bin Abdulaziz Hospital, Riyadh between May and August 2020. Data were obtained for the patient's demography, body mass index (BMI), and comorbidities. Additional data on patients that required intensive care unit (ICU) admission and clinical outcomes. A total of 565 positive patients (63 died and 101 required ICU). Univariate cox proportional hazards regression model showed that COVID-19 positive patients requiring ICU admission [Hazard's ratio, HR=4.2 95% confidence interval, CI 2.5– 7.2); p< 0.001] with preexisting cardiovascular [HR=4.1 (CI 2.5– 6.7); p< 0.001] or respiratory [HR=4.0 (CI 2.0– 8.1); p=0.010] diseases were at significantly higher risk for mortality among the positive patients.

Hu et al [35] developed a machine learning model for the early prediction of the mortality risk of COVID-19 patients. A data set of 183 patients (115 survivors and 68 non-survivors from COVID-19) from the Sino-French New City Branch of Tongji Hospital. Total of 64 patients (33 survivors and 31 non-survivors from COVID-19). Demographic, clinical, and first laboratory data after admission were extracted from patients' medical records. The study initially attempted 10 methods and then selected five of them (LR, partial least squares (PLS) regression, elastic net (EN) model, RF, and bagged flexible discriminant analysis FDA)) according to the model's performance and property to be reported. The LR model, RF, and bagged FDA yielded similar performance, as measured by the AUC. LR was selected as the final model because of its simplicity and high interpretability. The most essential four variables selected by the models were: age, hsCRP level, lymphocyte count, and D-dimer level. The performance of the model was evaluated using both 10-fold cross-validation on the training data set and independent testing using the external validation set. The AUC, sensitivity, and specificity reached 89.5%, 89.2%, and 68.7% during cross-validation and 88.1%, 83.9%, and

79.4% with independent testing, respectively. The study found that non-survivors were more likely to be male and older than survivors. Moreover, levels of all the inflammatory factors were higher in the non-survivors than in the survivors. In particular, levels of hsCRP and D-dimer were more than six times and almost three times higher in non-survivors than in survivors.

Zao Z. [12] Developed a risk-score model to predict mortality and ICU admission. The study used a data set from 641 laboratory-confirmed COVID-19 patients (195 admitted to the ICU, 82 expired) from Stony Brook University Hospital, USA. Symptoms, comorbidities, demographics, laboratory findings, vital signs, and imaging findings were all compared with those of non-critical COVID-19 patients to identify the most significant variables predicting the two outcomes. The study employed the ML approach and LR and achieved good accuracy with an AUC of 0.83 for mortality prediction and 0.74 for ICU admission prediction on the testing data set. The study found that the common top predictors of mortality and ICU admission were elevated LDH, procalcitonin, and reduced SpO2. Moreover, a reduced lymphocyte count and smoking history were among the top predictors of ICU admission but were not associated with increased mortality in this study. On the other hand, cardiopulmonary parameters (i.e., history of heart failure, chronic obstructive pulmonary disease (COPD), elevated heart rate) were among the top predictors of mortality in COVID-19 patients, but ICU admission was not.

Tschoellitsch et al. [52] developed a model using a Random Forest Machine learning algorithm to predict the diagnosis of COVID-19 based on patient blood tests. A data set of 1528 patients (65 positives) was employed to build the model, which achieved an accuracy of 81%, an area under the receiver operating characteristic curve (AUC) of 0.74, a sensitivity of 60%, and a specificity of 82%. The most important features in predicting diagnosis were: leukocyte count, red blood cell distribution width (RDW), haemoglobin, and serum calcium.

Zhao et al. [78] developed a risk-score model to predict mortality and ICU admission. The study used a data set from 641 laboratory-confirmed COVID-19 patients (195 admitted to the ICU, 82 expired) from Stony Brook University Hospital, USA. Symptoms, comorbidities, demographics, laboratory findings, vital signs, and imaging findings were all compared with those of non-critical COVID-19 patients to identify the most significant variables predicting the two outcomes. The study employed the ML approach and LR and achieved good accuracy with an AUC of 0.83 for mortality prediction and 0.74 for ICU admission prediction on the testing data set. The study found that the common top predictors of mortality and ICU admission were elevated LDH, procalcitonin, and reduced SpO2. Moreover, a reduced lymphocyte count and

smoking history were among the top predictors of ICU admission but were not associated with increased mortality in this study. On the other hand, cardiopulmonary parameters (i.e., history of heart failure, chronic obstructive pulmonary disease (COPD), elevated heart rate) were among the top predictors of mortality in COVID-19 patients, but ICU admission was not.

Yao et al. [75] developed a model to predict the severity of COVID-19 using blood or urine test data. The data set consisted of 137 patients (75 severely ill) from the Tongji Hospital Affiliated to Huazhong University of Science and Technology. The ML algorithm SVM was used to build the severeness detection model, which achieved an accuracy of 81.48%. The highest-ranking features detected by the model were age, blood test values (neutrophil percentage, calcium, and monocyte percentage), and urine test values (urine protein, red blood cells (occult), and pH (urine)).

Izquierdo et al. [90] developed a model to predict ICU admission using an ML data-driven algorithm. The study used a data set of 10,504 COVID-19 patients (1353 hospitalized, 83 admitted to ICU) from the general population of the region of Castilla-La Mancha (Spain), which included clinical information regarding the diagnosis, progression, and outcome of the infection. A DT algorithm was employed. The model achieved accuracy, recall, and AUC values of 0.68, 0.71, and 0.76, respectively. The three variables that contributed most to predicting ICU admission were age, fever, and tachypnea with or without respiratory crackles.

Li et al. [94] developed a model to predict the mortality of COVID-19. The ML algorithms GBDT, LR model, and simplified LR were trained and validated using a data set of 2924 patients including 257 non-survivors. The GBDT achieved the highest fivefold AUC of 0.941. The study found that leukomonocyte (%), urea, age, and SpO2 were the best predictors of mortality.

2.8. Summary of Related work

Table 2-3 Summery of related work

Author	**Title**	**Model**	**Description**	**Result**	**Gap**
Booth [36]	Development of a prognostic model for	LR and SVM	Amount of data used for this prediction Total 398aptients. Out	AUC 93%, 91% sensitivity, and 91% specificity	-The obtained accuracy is high, but it is not sufficient enough to

	mortality in covid-19 infection using machine learning		of this 355 patients are survivors and 43 patients are non-survivors.		prognostic the outcome of the disease. -The amount data size relatively small with other researches.
Shaikh Fs-et [34]	Comorbidities and Risk Factors for Severe Outcomes in COVID-19 Patients in Saudi Arabia: A Retrospective Cohort Study	Cox Model	A total of 565 COVID-19 positive patients were inducted in the study out of which, 63 (11.1%) patients died while 101 (17.9%) patients required ICU admission.	Hazard's ratio, HR=4.2 95% confidence interval, CI 2.5–7.2); p< 0.001] with preexisting cardiovascular [HR=4.1 (CI 2.5– 6.7); p< 0.001] or respiratory [HR=4.0 (CI 2.0-8.1); p=0.010	-The amount of data used for prognostic model relatively small. -Only cox model used for comparison.
Shoer et al [37]	A prediction model to prioritize individuals for sars-cov-2 test built from national symptom surveys	LR	A data collected from 498 COVID-19 positive patient symptoms	AUC 0.737	-No performance comparison with other models. - Small amount of data used for prognostic -Insufficient accuracy for the model.

Hu et al [35].	Prognostic factors for covid-19 pneumonia progression to severe symptoms based on earlier clinical features: a retrospective analysis	LR	A data set of 183 patients (115 survivors and 68 non-survivors from COVID-19) from the Sino-French New City Branch of Tongji Hospital. Total of 64 patients (33 survivors and 31 non-survivors from COVID-19).	AUC of 94.4%, sensitivity of 94.1%, and specificity of 90.2%.	-Small amount of data size used for the study. -No comparison with other models.
Izquierdo et al [38]	Clinical characteristics and prognostic factors for intensive care unit admission of patients with covid-19:	DT	The study used a data set of COVID-19 patients (1353 hospitalized, 83 admitted to ICU) from the general population of the region of Castilla-La Mancha (Spain),	AUC of 76%, accuracy 68%, and recall 71%	-Insufficient accuracy for prognostic model. -No comparison with other ML models.
Tschoellitsch et al. [39]	Machine learning prediction of sars-cov-2 polymerase chain reaction results with	RF	A data set of 1528 patients (65 positives) was employed to build the model, to predict the diagnosis of COVID-19 based	Accuracy of 81%, area under ROC curve of 0.74 sensitivity of 60%, and specificity of 65%.	-No performance comparison with other models. -Insufficient accuracy for the model.

	routine blood tests		on patient blood tests		

A review is presented in this chapter relating to the application of machine learning to identify the clinically relevant prognostic factors for the prognostic of covid-19 outcome leading to the mortality risk. Initially, literature related to the prognostic model are studied to gain knowledge about the model development. In a machine learning section highlighting is given five supervised machine learning techniques, Logistic Regression (LR) and Support Vector Machine (SVM), Multiline Perceptron (MP), and K-nearest neighbour (KNN). Which are applied in the area of medical sciences

CHAPTER THREE

3. RESEARCH METHODOLOGY

In this chapter, the research methodology to build the datasets and techniques to achieve research objectives and answer the research question are discussed. This chapter explains and justifies the methodologies used in conducting the study on building a prognostic model for COVID-19 outcomes.

3.1. General Approach

The methodological analysis of prognosis covid-19 is shown in Figure.3.1 and it includes the following steps, such as:

- Review literature on previous studies made on prognosis covid-19 using traditional machine learning techniques.
- Collect patient clinical, demographic and laboratory data.
- Preparing the necessary data set by using preprocessing, EDA.
- Design a machine learning model.
- Train the proposed model by using the collected dataset.
- Test the proposed model with a test dataset.
- Evaluate the model by using performance metrics.

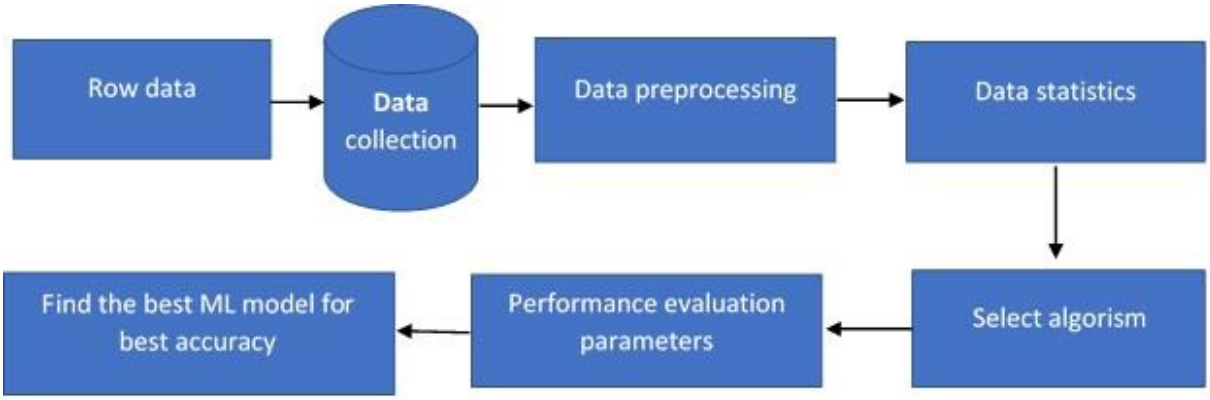

Figure 3-1 Block diagram of research flow

3.2. Literature Review

By searching different related literature from the internet (Books, Journals, etc.) will be reviewed to understand the concept of prognostic analysis and machine learning how they applied to solve the related problem in the prior research. To achieve this research objective the most recent and current literature will be reviewed.

Literature for this review was identified by searching the following online databases: BioRxiv, MedRxiv, ChemRxiv, Google Scholar, and PubMed. These online databases contain archives of most English biomedical journals. During the revision of this literature, the gap of the previous solution will be identified to use input for this proposed solution. This is the important and necessary content where all possible reference and journal related to research is investigated and analyzed.

3.3. Material and Tools

3.3.1. Software Tools

This section gives a brief overview of the tools used in the project and the implementation of the solution. It is included in the study for others to be able to reproduce the study using the same approach, as the code written in the study is not public due to using sensitive data. The implementation of the experiments used in the project is written in the programming language Python. It also uses multiple supplementary libraries such as; Pandas, Numpy, Scikit-Learn, and Keras. The most used libraries are explained below.

Anaconda: is used for the implementation of the model and it is free and open-source distribution of the python and R programming languages for image processing, data science, machine learning, and related applications that aim to add and simplify package management and deployment. It contains different IDE's which are used to write the coding part such as Qt Console, Jupyter Notebook, Visual studio, and Spyder. We have used the Jupyter notebook and visual studio to implement the coding part. It is easy to use and run in a web browser.

TensorFlow: is a free and open-source library developed by Google and it is currently the most famous and fastest AI library. It can be used in any desktop which runs Windows, macOS, Linux, in the cloud as a service and mobile devices like iOS and Android. The architecture of TensorFlow works for the preprocessing of the data, building the model, train the model, and estimate the

model. All the Computation in TensorFlow involves tensors (n-dimension array) that represent all kinds of data [40].

Keras: is a high-level API for writing Neural Networks which can run on top of TensorFlow. Keras focuses on user-friendliness, modularity, and easy extensibility. It simplified the process of creating a neural network, which made it possible to spend more time on data processing and the neural networks architecture and feature engineering in this project. All models created in the study are created using Keras [41].

Scikit-learn: Scikit-learn is a library offering various kinds of ML algorithms such as SVMs, KNN and Decision Tree. It also provides multiples tools like dimensionality reduction, model selection and preprocessing to prepare and tune data for algorithms [42] (Buitinck, 2013).

Pandas: It is a python package that provides expressive data structures designed to work with both relational and labelled data. It is an open-source python library that allows reading and writing data between data structures.

3.3.2. Hardware Tools

To implement the machine learning algorithm with the selected software tools I will use HP pavilion laptop with the following specification-CPU Intel(R) Core(TM) i7-7500 CPU @ 2.70GHz processor, with in 8GB RAM.

3.4. Data collection

Data collection was an essential and delayed process. Regardless of the field of research, the accuracy of the data collection is essential to maintain structure. As the clinical information of patients was not publicly available, it was an inflexible and tedious process to collect the data. Various Hospitals and Health Institutes in Ethiopia were approached to get the most accurate data but due to the present situation at hospitals with heavy inflow of patients with COVID-19, we couldn't get access to direct information. An intense search was conducted on various hospitals to gather patient clinical data.

3.5. Dataset Preparation

The data set that was used to train the model to predict COVID-19 was gathered from SPHMMC and Adama Hospital. The data set contained information about hospitalized patients with COVID-19. To protect the privacy of patients each case are labelled with ID noted in the dataset, which is

an actual individual. This file contains demographic data, symptoms, previous medical records, laboratory values that were extracted from electronic records and form paper.

3.6. Data Pre-processing

The data preprocessing phase is responsible for constructing the final dataset that will be deployed for learning and constructing the ML model. The data preprocessing phase covers all the activities involved in the transformation of preprocessing. It consists of steps like data cleaning and data transformation phases.

Additionally, it is important to split the initial data into two mutually independent datasets namely the training dataset and testing dataset during this phase to propose a reliable approach for the estimation of the true performance of the constructed Prognostic models. The training dataset is used for the construction of the final Prognostic model while the validation dataset is used to test the constructed model developed using the training dataset [43].

3.6.1. Data cleaning

Data Cleaning means the process of identifying the incorrect, incomplete, inaccurate, irrelevant, missing part of the data, outlier, duplicate rows and then modifying, replacing or deleting them according to the necessity [44].

Missing value analysis: Real-world data would certainly have missing values. This could be due to many reasons such as data entry errors or data collection problems. Irrespective of the reasons, it is important to handle missing data because any statistical results based on a dataset with non-random missing values could be biased. There are many imputation techniques to deal with missing data but Multiple imputations (MI) has become a very popular tool for dealing with missing data in recent years for this research we used the multiple imputation method [45].

Outlier analysis: outlier indicates a data point that is significantly different from the other data points in the data set. Outliers can be created due to errors in the experiments or the variability in the measurements. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models, and, ultimately, more unexceptional results [46].

3.6.2. Data transformation

3.6.2.1. Scaling

The input values are rescaled to a uniform scale. To ensure that all the feature values are on the same scale, normalization or standardization is a mandatory step to be carried out before proceeding to a model building [47].

3.6.2.2. Balanced dataset

The main concern is that the data is highly imbalanced and small in size. There are approx. 20% of records have mortality risk as '1' and the rest 80% of records have mortality risk as '0'. If the experiment is proceeding further to the modelling phase without balancing the data then the model will be trained with biasing and the cost of misclassifying minority classes could be very high. Sampling techniques: under-sampling or over-sampling should be implemented to get rid of imbalanced data set. Since the data set is quite small in size, each instance is highly important and can't risk losing any information. [48]

3.7. Dimensionality Reduction

In machine learning and statistics, dimensionality reduction is another class of data transformation, which can reduce the number of variables by introducing a smaller number of variables but still owns more or less variation in the original variables. Classified into two types: feature selection and feature extraction.

3.7.1. Feature Extraction

After the data acquisition, all features of the EHR system are available, and that feature set can be huge. It is sometimes necessary to reduce the number of features and increase the information content of the features to increase accuracy, and to reduce the computational effort of a machine learning approach. Feature extraction and feature selection can help in performing this. Feature selection selects a subset of the most representative features, whereas feature extraction transforms the original feature space and receive new information by combining features.

3.7.2. Feature Selection

Feature selection is a common and useful technique to reduce computational costs and increase accuracy. Feature selection selects a subset of features that are defined on a representative basis of the different techniques that will be discussed in the following, no technique selects features.

3.8. Models

The main aim of this study is to investigate the use of an SVM, LR, RF, MLP and KNN based classification model for determining the prognostic clinical outcome. Therefore, five supervised machine learning algorithms, Multiline perceptron, Logistic Regression and Support Vector Machine, K-Nearest Neighbor, and Random Forest will be implemented to build the models. Algorithms are selected because of their good regression performance, and their popularity in solving prediction problems on previous research work results in related work. The following five machine learning algorithms are used to build prediction models. [36] [39] [12] [13]

3.9. Evaluation Metrics to Evaluate Accuracy of Model

This experiment uses the F1 score and the accuracy to evaluate the performance of the model. The model is trained on 70% of the data is used for 30% for testing. The metrics used to evaluate the model in this classification task are

- Accuracy or classification accuracy (CA)
- Precision
- Recall
- F1 score

Accuracy: - of the model will be calculated as the percentage of correct prediction of the top class and the target class assigned by the author beforehand are the same. It can be represented by the below

$$\text{Accuracv} = \frac{TP+TN}{TP+FP+FN+TN}$$

Equation 3-1 Accuracy

Where, TP (True positive) represents the positive instances that are correctly classified as positive, TN (True Negative) represents the negative instance that are correctly classified as negative, FP (False positive) represent the negative instance that are wrongly classified as positive, FN (False Negative) represents the positive that are wrongly classified as negative.

Precision: - is calculated as the fraction of True positives (TP) from the sum of the relevant classes, i.e. the sum of the True positives and the false positives. It can be represented by the below

$$\text{Precision} = \frac{TP}{TP+FN}$$

Equation 3-2 Precision

Recall: - is calculated as the fraction of True positives from the sum of True positives and False Negatives. It can be represented by the below

$$\text{Recall} = \frac{TP}{TP+FN}$$

Equation 3-3 Recall

F1 score:-will be used in this experiment as the dataset is imbalanced. F1 score is interpreted as the harmonic mean of precision and Recall. It can be represented by the below

$$\text{F1 Score} = \frac{2*\text{precession}*\text{Recall}}{\text{Precession} + \text{Recall}}$$

Equation 3-4 F1-score

A confusion matrix: - is also an evaluation metric that is used to describe the performance of a classification task. Precision, Recall and Accuracy can all be calculated with the help of a confusion matrix

Table 3-1 Confusion matrix

	Positive (Actual)	Negative (Actual)
Positive (prediction)	True positive(TP)	False Positive (FP)
Negative (prediction)	False Negative (FN)	True Negative (TN

CHAPTER FOUR

4. PROPOSED WORK

This chapter focuses on the design of the proposed work. The layout of the chapter is a mirror of the previous chapter 'Methodology part' so that a comparison can be made between the proposed models to build a prognostic covid-19 outcome using machine learning techniques.

4.1. The proposed model to build a prognostic covid-19 outcome

The proposed solution is based on the framework shown in Figure 4-1. It takes COVID-19 clinical, laboratory and demographic data as input. After collecting the data, models are developed using KNN, MLP, SVM, LR and RF machine learning algorithms and trained on a training set of the whole dataset. The models are evaluated using accuracy, f1-score, precision, and recall. The evaluation results were used to select the best prognostic model. The product of these tasks is to build a model used for prognostic COVID-19 outcome.

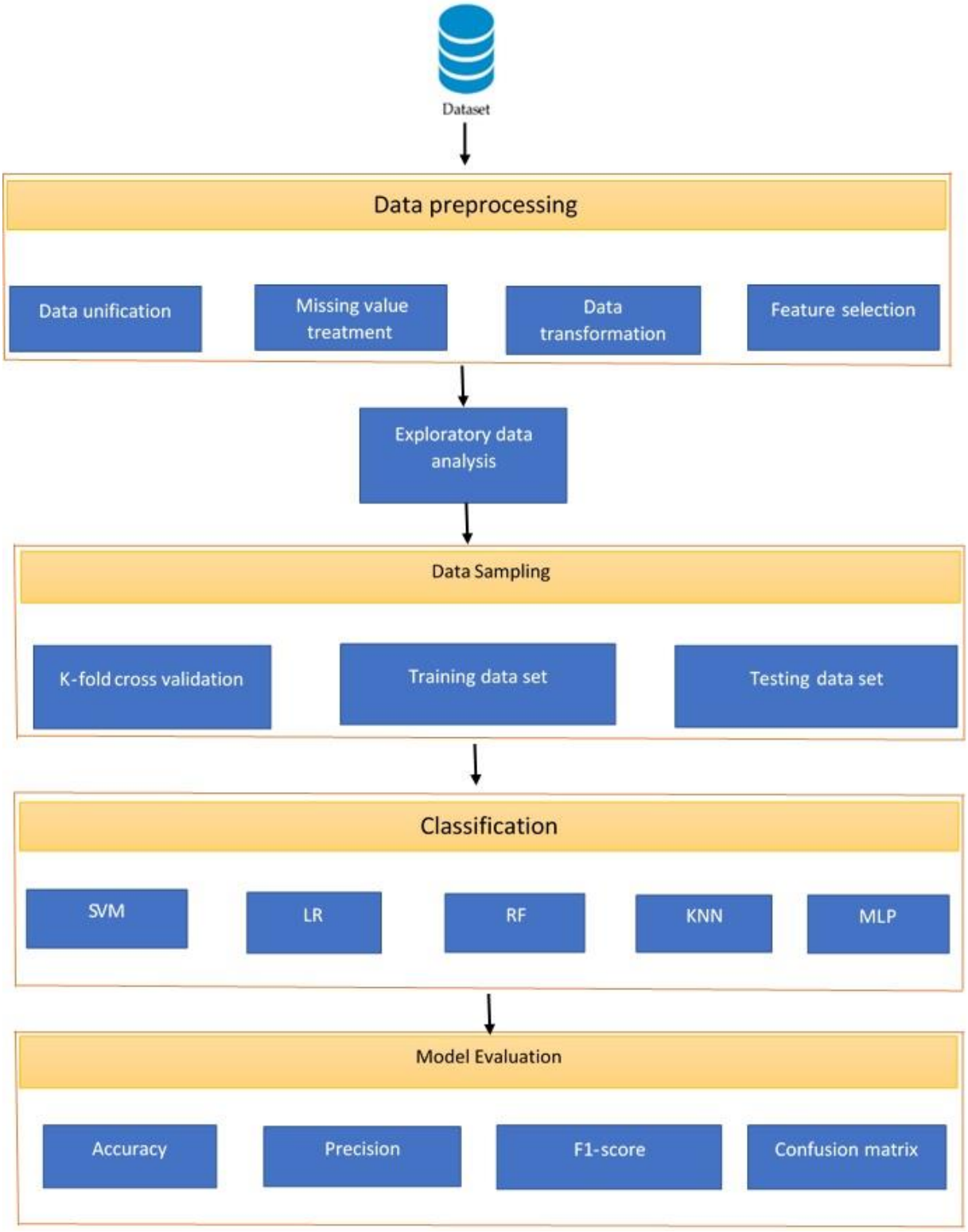

Figure 4-1 Block diagram of the proposed work.

4.2. Dataset Description

In this study, one available epidemiological dataset was obtained processed and used for analysis. Each case in the data represents an individual who tested positive for COVID-19 gathered from SPHMMC and Adama Hospital. This data originally contains 1800 cases. To protect the privacy of patients, each case de individual and anonymized. The cases are labelled with ID noted in the dataset. The file contains variables, including ID, age, sex, comorbidity, and laboratory findings.

Table 4-1 Data description

NO	Attribute	Description
1	Age:	Age of the patient
2	BMI	Body mass index (BMI) is a measure of body fat based on height and weight that applies to adult men and women.
3	Cholesterol	Cholesterol is a type of body fat, or lipid. A serum cholesterol level is a measurement of certain elements in the blood, including the amount of high- and low-density lipoprotein cholesterol (HDL and LDL) in a person's blood.
4	Bp(thalach)	resting blood pressure (in mm Hg on admission to the hospital)
5	Heart rate	maximum heart rate achieved
6	Sex	1-Women 2-Male
7	Cough	0- did not have cough 1- had a high cough
8	High Fever	0- did not have fever 1- Had a high fever
9	Weakness/pain	0- had no weakness or pain 1- Felt weakness or pain
10	Cardiac	0- did not display cardiac related symptoms 1-display cardiac related symptoms

11	Nausea	0- did not experience nausea 1- Experienced nausea
12	Kidney S	0- don't display kidney related disease 1- Does display kidney related disease
13	Diabetic	0- Does not display diabetes 1- Does have diabetes
14	Hypertension	0- does not have hypertension 1- Does have hypertension
15	Cancer	0-does not have cancer 1- Does have cancer
16	Patient outcome	1-alive 0-dead

Clinical Outcomes

The primary outcome was defined as patient mortality. The secondary outcome was surviving from the virus. Risk factors associated with outcomes were analyzed and compared between initially asymptomatic and symptomatic patients. We established a prediction model for patient mortality through risk factor analysis among initially symptomatic patients.

4.3. Descriptive analyses

We performed descriptive analyses of the predictors by respective stratification groups and present the results as numbers and proportions. Potential correlations between predictors were tested with Pearson's correlation coefficient.

4.4. Preprocessing

Preprocessing raw data is one of the first steps in building a machine learning model. Here, preprocessing consists of several subtasks like imputation, scaling, normalization, feature selection and feature extraction. Data pre-processing is always needed during the implementation of machine learning algorithms, since different models have different requirements to the predictors in the mode, and different data preparation can give rise to different predictive performances. The cross-validated resampling technique can be often used to evaluate the model generalizability,

where a training set is used to fit a model and the testing set is used to estimate the efficacy.

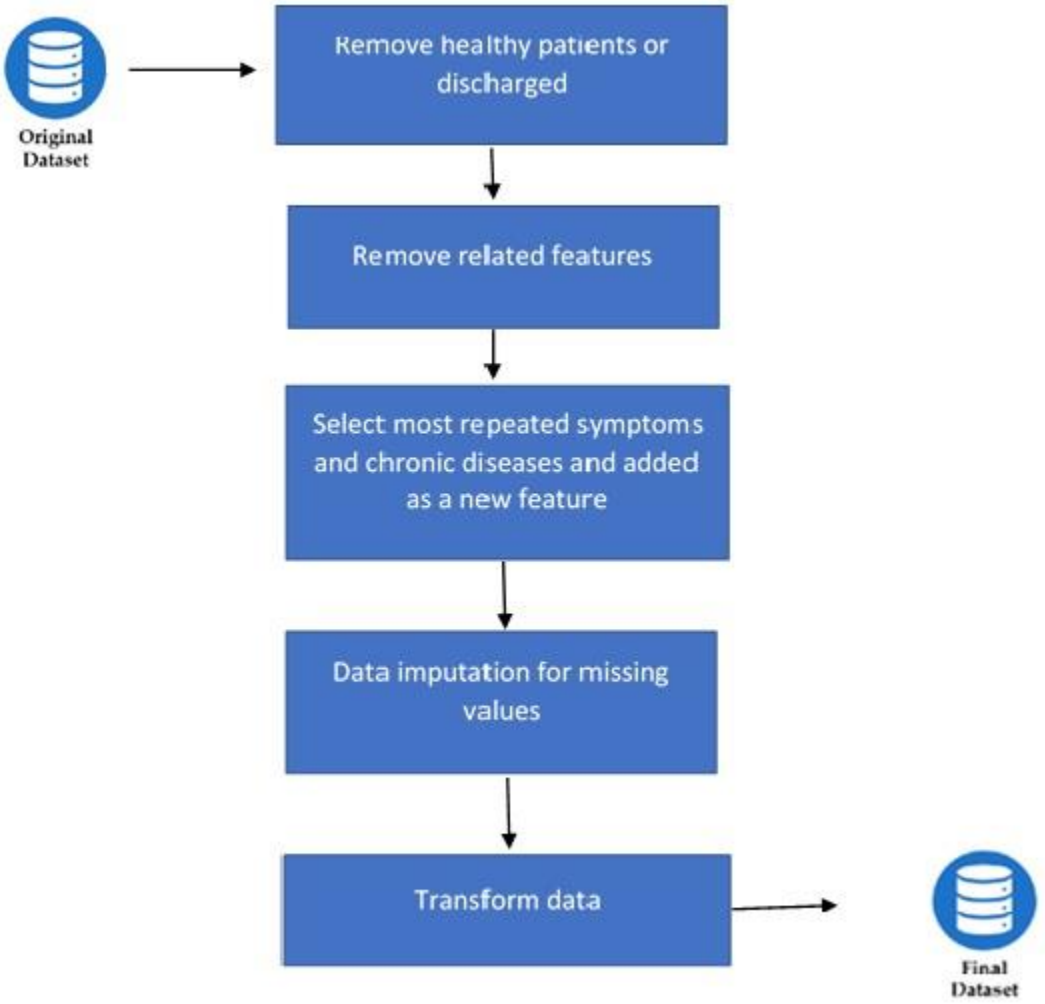

Figure 4-2 Block diagram of preprocessing steps

4.4.1. Data Cleaning

Imputation: - via chained equations to impute the missing data is an iterative method in which multiple values are estimated for the missing data points using the distribution of the observed data. The algorithm in two variations to impute categorical and numeric data. The advantage of this method is that it reflects the uncertainty around the true value and returns unbiased estimates [45].

Outlier Analysis

Z-Score

Z-score is a statistical measure that tells you how far is a data point from the rest of the dataset. In a more technical term, Z-score tells how many standard deviations away a given observation is from the mean. Some Python libraries like Scipy and Sci-kit Learn have easy to use functions and classes for easy implementation along with Pandas and Numpy [46].

4.4.2. Data Split

Results from the executions of the algorithms need to be evaluated to see which ones are better and also to see if the parameters' values used are acceptable. As only one labelled dataset is provided, it has to be split into 2 smaller parts to train, test and validate the algorithms. For this purpose, then, the main (train) database will be split into train, test subsets. This division will be done in the following proportions: 80% train, 20 test.

4.4.3. Data Transformation

4.4.3.1. Standardizing and Normalization

Normalization: - The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. This step helps to accurately estimate the minimum and maximum observable values. Standardizing: - Standardization is a pre-processing step to standardize values of features from different dynamic ranges into a specific range. Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

4.5. Dimensional Reduction

4.5.1. Feature selection

Correlation analysis: Correlation analysis is one of the most common techniques to select features. This approach evaluates a linear relationship between pair-wise inputs with a correlation function [38], which can remove features with redundant behaviours. This technique, however, fails if the amount of sample data is low, or if the relationship between features is non-linear.

4.6. Models

In this stage, classification models are built to predict the mortality risk of the patient using LR, KNN, MLP, RF and SVM algorithms. Before training the model, input data is first normalized.

Also, correlation analysis is performed for the better fit of the model. All the variables are very weakly correlated with each other and hence none are dropped while training the model.

4.6.1. Support vector machine (SVM)

SVM is a supervised machine learning algorithm that can be used for both classification and regression. SVM is transforming a training data set into a higher dimension, it optimizes a hyperplane that separates the two classes with minimum classification errors.

4.6.2. Random forest (RF)

RF is a DT ensemble method for classification tasks output of random forest is selected by most tress. For regression tasks, the mean or average prediction of the individual trees is returned. That creates multiple trees through a re-sampling process called bagging. Numerous DTs are constructed by re-sampling using bagging with replacement. Each node of the tree is split using a subset of the attributes that are selected randomly for each tree. Class membership for a new example is identified as the most commonly predicted class from the DTs by a simple unweighted majority vote [26].

4.6.3. K-nearest-neighbor (KNN)

One of the simplest Machine Learning algorithms is based on the Supervised Learning technique. KNN is a classifier that learns by comparing a given unlabeled data point with the training data set. It searches for the K most similar data points, referred to as the KNNs. A distance metric, such as Euclidean distance, is usually used to measure closeness. The algorithm then finds the most common class among its KNNs and assigns it to the given data point [28].

4.6.4. Logistic regression (LR)

LR models the probability of data points belonging to a certain class based on the value of independent features. It then uses the model to predict the probability that a given data point belongs to a certain class. Usually, the sigmoid function is used in building the regression model. It is assumed that the data points follow a linear function.

4.7. K-Fold Cross-Validation

K-Fold Cross-validation is a technique used in machine learning to help figure out good settings for hyper parameters of a model. The technique is used to prevent bias and variance. In K-Fold cross-validation, we split our training set into K number of subsets, called folds. The model is then trained k times, each time one of the k subsets is used for validation, and the other subsets are used

for training. At the end of the training, the performances of the model on each fold is averaged to come up with the final validation metrics for the model. For hyper parameter tuning, the Cross-validation process is repeated several times, with each iteration using different model settings. The best model is chosen and trained on all the training data, and later evaluated on the test data [26].

4.8. Models Evaluation and Testing

To test and estimate the learning ability of machine learning models trained on the COVID-19 cases dataset. The basic concern of the machine learning model is to find an accurate estimation of the generalization error of the trained models on finite datasets. Because of this reason this study used different performance evaluation metrics appropriate for models, these metrics are Accuracy, Precision, F1-score and confusion matrix.

CHAPTER FIVE

5. IMPLEMENTATION

5.1. Introduction

The implementation phase describes building a prognostic model for COVID-19 outcome. By using the methodology discussed in chapter three. Also, implement the proposed solution in chapter four. This study follows an experimental approach to determine the best machine learning algorithm for building a prognostic model for COVID-19 outcome. Moreover, performs experiments by using the dataset.

5.2. Data and Environment Setup

The research was implemented using python 3.7.7 using Jupyter notebook as the main integrated development environment (IDE). Python language was selected as there is a lot of support from an active community for prediction using Tensorflow with Keras.

Import and store ML models

To build a machine learning model using the proposed algorithm, we used a python ML library packages. To build a machine learning algorithm we used five different models from five different categories of machine learning algorithms

```
import re
import itertools
import warnings
import matplotlib.pyplot as plt
import tensorflow as tf
import seaborn as sns
import pandas as pd
import numpy as np
from tensorflow import keras
```

```
# Import plotly tools
import plotly.offline as py
import plotly.graph_objs as go
import plotly.tools as tls
```

```
# Import keras tools
from tensorflow.keras import regularizers
from tensorflow.keras.callbacks import History
from tensorflow.keras.layers import Dense, Input, Dropout
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
```

```
# Import other tools
from __future__ import print_function
from pandas import read_excel
from IPython.display import Image
from collections import Counter
from itertools import cycle
from scipy import stats, integrate, interp
from subprocess import check_output
import os
files = os.listdir('./')
```

```
# Import relevant machine learning models
import sklearn
import lightgbm as lgb # Speed
#Hyperparameter
from sklearn import decomposition, preprocessing, svm
# Ensemble
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor,VotingClassifier, ExtraTreesClassifier
# Regression
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.multiclass import OneVsRestClassifier
# Instance Based
from sklearn.neighbors import KNeighborsClassifier
# Nueral Network
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
```

Figure 5-1Import data preprocessing and analysis tools

```
# Input data files are available in the "../input/" directory
dataset = pd.read_csv("C:/Users/EYASU/Desktop/experiment/dataset.csv", na_values="?", low_memory = False)
```

Figure 5-2 Import dataset

5.3. Descriptive analysis

The frequency plot of categorical variables; weakness/pain, fever, nausea, cardiac, high fever, kidney, diabetes, hypertension, cancer, death/alive is plotted as shown in figure 5.3 The ‘Target’ variable is highly biased as per the information provided by the bar graph. Only 20% of the values are ’1’ and rest of the records have ’0’ values. The number of patients having high mortality risk is almost (1/4) th of the number of patients without any mortality risk. The balancing of the target feature will be taken care of in Data Preparation section. All the categorical variables are binary and have ’Yes’ or ’No’ values.

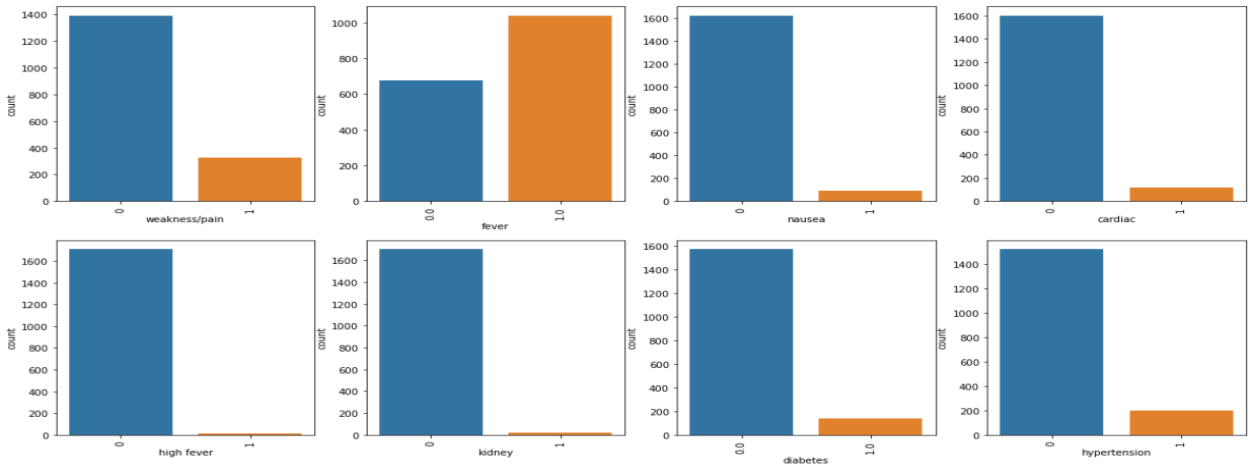

Figure 5-3 Distribution plot of categorical features with the target

The distribution of the numeric features concerning the target variable. Age, Serum Cholesterol, Systolic BP, BMI, Pulse pressure are normally distributed with the mortality risk.

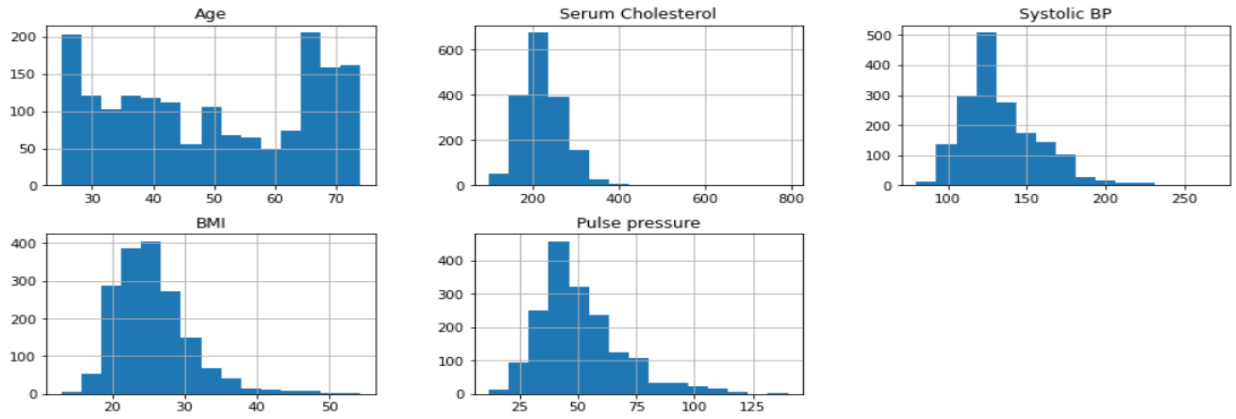

Figure 5-4 Distribution plot of numeric features with the target

The below figure shows information about the mean, standard deviation, maximum value, minimum value and distribution (quartile range) of each numeric variable.

	Age	Serum Cholesterol	Systolic BP	BMI	Pulse pressure
count	1715.000000	1715.000000	1716.000000	1714.000000	1716.000000
mean	49.413411	222.562799	133.660256	25.533655	50.903263
std	16.038713	50.438335	24.760131	5.217761	17.925989
min	25.000000	98.000000	80.000000	12.941206	12.000000
25%	35.000000	188.000000	116.000000	21.944504	38.000000
50%	49.000000	217.000000	130.000000	24.919341	48.000000
75%	66.000000	251.000000	148.000000	27.973385	60.000000
max	74.000000	793.000000	270.000000	54.336876	140.000000

Figure 5-5 Statistics table

5.4. Data Preprocessing

5.4.1. Imputation

```
#Imputation
dataset.isnull().sum()
```

Figure 5-6 Missing value

only five variables, 'Age', 'Fever',' Serum Cholesterol', and 'BMI' has missing value out of 16 variables and is given in table 5.1.

Table 5-1 Missed values

Variable	Missing Count
Age	1
Fever	1
Diabetes	1
Serum Cholesterol	1
BMI	2

5.4.2. Scaling

```
scaler = StandardScaler()
X = scaler.fit_transform(X)

features_all = dataset[dataset.columns[:13]]
features_standard = scaler.fit_transform(dataset[dataset.columns[:13]]) #
X = pd.DataFrame(features_standard,columns=[feature13])
['pred_attribute'] = dataset['pred_attribute']
outcome = X['pred_attribute']
```

Figure 5-7 Scaling

5.4.3. Data Split

When we split the dataset into train and test datasets, the split is completely random. Thus the instances of each class label or outcome in the train or test datasets is random. Thus we may have many instances of class 1 in training data and less instances of class 2 in the training data. So during classification, we may have accurate predictions for class1 but not for class2. Thus we stratify the data, so that we have proportionate data for all the classes in both the training and testing data.

Table 5-2 Data Split with Balanced data

	Training dataset	Test Dataset	Total
Alive	508	233	766
Death	508	233	766

5.5. Feature Selection

5.5.1. Correlation

Pearson correlation coefficient is used in the experiment to interpret the linear association between the numeric-continuous variables. The correlation coefficient ranges between -1 to 1, the greater the absolute value the stronger the linear relationship. The correlation heatmap matrix as shown in figure 4.5, gives the strength of the relationship between the features. The result deduced from the matrix is as under:

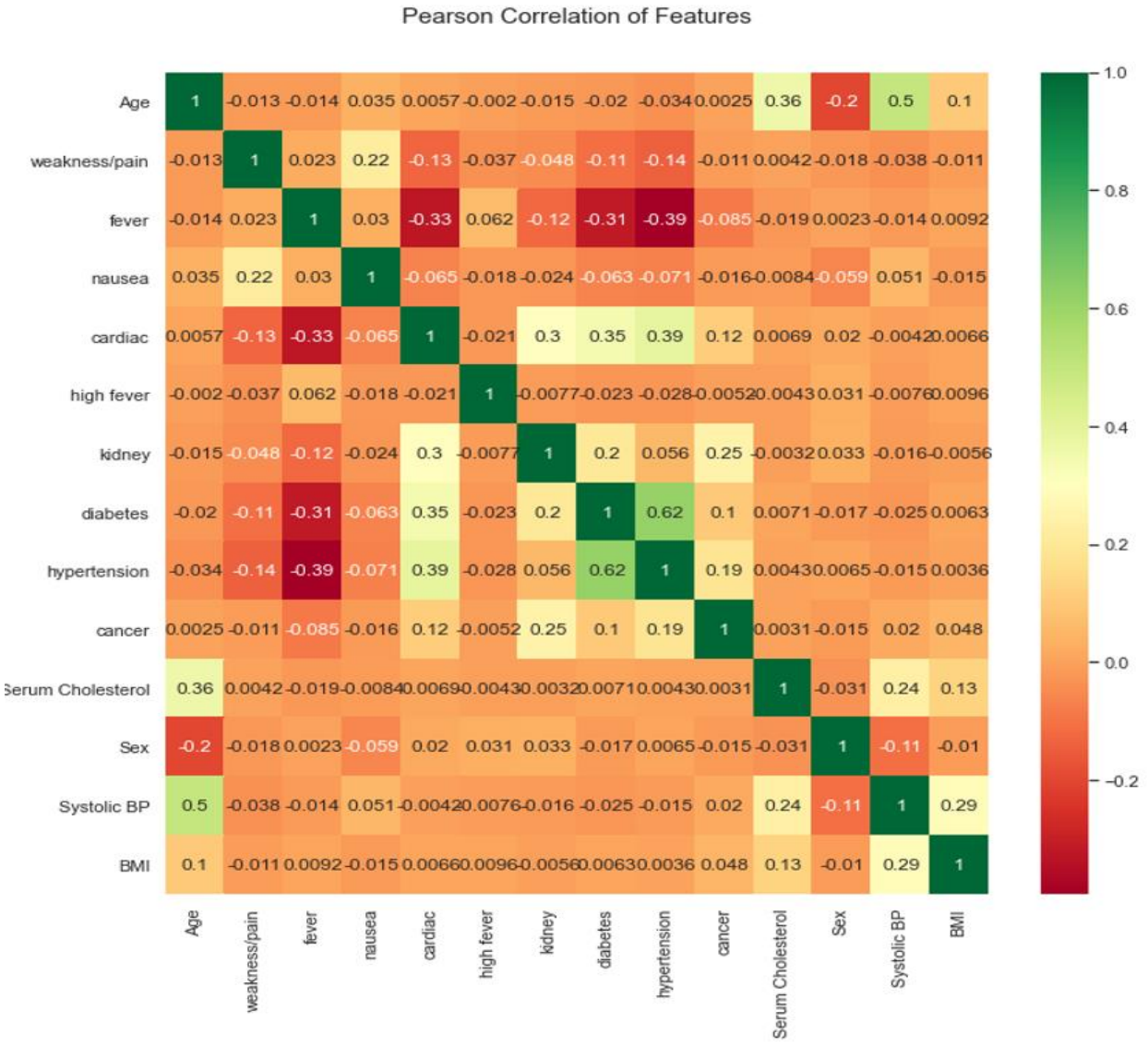

Figure 5-8 Correlation

The Pearson Correlation Heatmap plot indicated that there are no strongly correlated features. This is good from a point of view of feeding these features into the learning model because this means that there isn't much redundant or superfluous data in our training set. No features can select from this plot.

5.5.2. Feature importance

The feature importance (variable importance) describes which features are relevant. It can help with a better understanding of the solved problem and sometimes lead to model improvements by employing the feature selection [49].

5.5.2.1. Random Forest Classifier

The Random Forest algorithm has built-in feature importance which can be computed using Gini-importance.

```
model = RandomForestClassifier(n_estimators=100,random_state=random_state)
model.fit(X, y)
pd.Series(model.feature_importances_, index=data.columns).sort_values(ascending=False)
```

Figure 5-9 Radom forest classifier

5.5.2.2. Extra Tress classifier

This algorithm is very similar to Random Forest, extra tree classifier has built-in feature importance.

```
model = ExtraTreesClassifier(n_estimators=100,random_state=random_state)
model.fit(X, y)
pd.Series(model.feature_importances_, index=data.columns).sort_values(ascending=False)
```

Figure 5-10 Extra feature classifier

5.6. Model Building and Evaluation

In this stage, classification models are built to the prognostic covid-19 outcome of the disease using LR, SVM, KNN, RF and MLP algorithms. Before training the model, input data is first normalized. Also, correlation analysis is performed for the better fit of the model. All the variables are very weakly correlated with each other and hence none are dropped while training the model. In total, five supervised machine learning models are built.

```
# Initial tool settings
%matplotlib inline
warnings.filterwarnings('ignore')
warnings.filterwarnings('ignore', category=DeprecationWarning)
plt.style.use('fivethirtyeight')
sns.set(style='white', context='notebook', palette='deep')
py.init_notebook_mode(connected=True)


random_state = 43

names = ["k-Nearest Neighbors",
         "Support Vector Machine",
         "Random Forest",
         "Logistic Regression",
         "Multilayer Perceptron",
        ]

classifiers = {  "k-Nearest Neighbors" : KNeighborsClassifier(n_neighbors=3),
                 "Support Vector Machine" :  SVC(random_state=random_state),
                 "Random Forest" : RandomForestClassifier(random_state=random_state),
                 "Logistic Regression" : LogisticRegression(random_state=random_state),
                 "Multilayer Perceptron" :  MLPClassifier(hidden_layer_sizes=(100,),momentum=0.9,solver='sgd',random_state=ra
                }

algorithms = [ KNeighborsClassifier(n_neighbors=3),
               SVC(random_state=random_state),
               RandomForestClassifier(random_state=random_state),
               LogisticRegression(random_state=random_state),
               MLPClassifier(hidden_layer_sizes=(100,),momentum=0.9,solver='sgd',random_state=random_state),
                ]
```

Figure 5.11 Machine Learning models

5.6.1. Hyperparameter Optimization

Hyperparameters are important for machine learning algorithms since they directly control the behaviours of training algorithms and have a significant effect on the performance of machine learning models. Selecting the best possible parameters for our models and improve a model's performance by tuning its parameters.

5.6.1.1. SVM

SVM popular machine learning used for classification algorithms. For this sake, the best set of parameters are found by a process called grid search methods. Grid search iterates through all the possible combinations to find the best set of parameters.

```
svc = SVC(probability=True, random_state=random_state)
# Set the parameters by cross-validation
svc_pg = [{'kernel': ['rbf'], 'gamma': [1e-1, 1e-2, 1e-3, 1e-4],'C': [1, 10, 100, 1000]},
          {'kernel': ['linear'], 'C': [1, 10, 100, 1000]}]

scores = ['precision', 'recall']

for score in scores:
    svc.fit(X_train, y_train)
    y_eval = svc.predict(X_test)
    acc = sum(y_eval == y_test) / float(len(y_test))
    print("Accuracy of SVC: %.2f%%" % (100*acc))

    print("Tuning hyper-parameters for %s\n" % score)
    svc_gscv = GridSearchCV(svc, svc_pg, cv=kfold, scoring='%s_macro' % score)
    svc_gscv.fit(X_train, y_train)
    print("Best parameters set found on training set:")
    print(svc_gscv.best_params_,"\n")
    print("Grid scores on training set:")
    means = svc_gscv.cv_results_['mean_test_score']
    stds = svc_gscv.cv_results_['std_test_score']

    for mean, std, params in zip(means, stds, svc_gscv.cv_results_['params']):
        print("%0.3f (+/-%0.03f) for %r" % (mean, std * 2, params))
    print("Detailed classification report:")
    print("The model is trained on the full training set.")
    print("The scores are computed on the full test set.")
    y_true, y_pred = y_test, svc_gscv.predict(X_test)
    print(classification_report(y_true, y_pred))

    svc_est = svc_gscv.best_estimator_
    svc_score = svc_gscv.best_score_
    print("Best estimator for parameter C: %f\n" % (svc_est.C))
    print("Best score: %0.2f%%\n" % (100*svc_score))
```

Figure 5-11 SVM tuning model

5.6.1.2. KNN

This algorithm on of the simplest classification model. Even with such simplicity, it can give highly competitive results. KNN can give Accuracies for different values of nearest neighbours By using KNeighborsClassifier() method.

```
k_range=list(range(1,50))
data_copy=pd.Series()
k_scores = []
#x=[0,1,2,3,4]
for i in k_range:
    model=KNeighborsClassifier(n_neighbors=i)
    scores = cross_val_score(model, X, y, cv=kfold, scoring='accuracy')
    k_scores.append(scores.mean())
    model.fit(X_train,y_train)
    pred_test=model.predict(X_test)
    data_copy=data_copy.append(pd.Series(metrics.accuracy_score(pred_test,y_test)))

#plt.plot(k_range, k_scores)
plt.plot(k_range, data_copy)
plt.xlabel('Value of K for KNN')
plt.ylabel('Cross-Validated Accuracy')
#plt.xticks(X)
plt.show()
print('Accuracies for different values of n are:\n', data_copy.values)
```

Figure 5-12 K-NN model

5.6.1.3. Logistic Regression

It's like linear regression, Logistic regression is the right algorithm to work with classification algorithms. For this model we use GridSearchCv() method, this used to find the optimal hyper parameters of a model which results in the most 'accurate' predictions.

```
lr = LogisticRegression(
    C=0.1,
    penalty='l2',
    dual=False,
    tol=0.0001,
    fit_intercept=False,
    intercept_scaling=1.0,
    class_weight=None,
    random_state=random_state)

lr_pg = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}

lr_gscv = GridSearchCV(lr,param_grid=lr_pg, cv=kfold, scoring="accuracy", n_jobs= -1, verbose = 1)
lr_gscv.fit(X_train, y_train)

lr_est = lr_gscv.best_estimator_
lr_score = lr_gscv.best_score_
print("Best estimator:", lr_est, "\nBest Score:", lr_score)
```

Figure 5-13 LR tuning model

5.6.1.4. MLP

To implement the MLP model, the MLPClassifier() function is defined which has training data, batch size, epoch, and the number of neuron parameters. It uses relu (rectified linear unit) activation function. For this model we use GridSearchCv() method, this used to find the optimal hyperparameters of a model which results in the most 'accurate' predictions

```
mlp = MLPClassifier(momentum=0.15,solver='sgd',learning_rate_init=1.0, early_stopping=True, shuffle=True,random_state=random_

mlp_pg={
'learning_rate': ["constant", "invscaling", "adaptive"],
'hidden_layer_sizes': [(10,10),(100,),(100,10)],
#'hidden_layer_sizes': [x for x in itertools.product((10,20,30,40,50,100),repeat=3)],
#'tol': [1e-2, 1e-3, 1e-4],
#'epsilon': [1e-3, 1e-7, 1e-8],
'alpha': [1e-2, 1e-3, 1e-4],
#'activation': ["Logistic", "relu", "Tanh"]
}

mlp_gscv = GridSearchCV(mlp,param_grid=mlp_pg, cv=kfold, scoring="accuracy", n_jobs= -1, verbose = 1)
mlp_gscv.fit(X_train, y_train)

mlp_est = mlp_gscv.best_estimator_
mlp_score = mlp_gscv.best_score_
print("Best estimator:", mlp_est, "\nBest Score:", mlp_score)
```

Figure 5-14 MLP tuning model

CHAPTER SIX

6. RESULTS AND DISCUSSION

Introduction

This chapter describes the implementation of the prognosis model, which was specified in detail in the previous chapter. In this chapter, all the experimentation details such as the results of each experiment and discussion of these results are presented briefly. The result of the experiment is shown in different figures and tables. In this section, the results of conducting this study are presented. In this section, the results are gained using only EHR data and the best model is identified.

6.1. Machine learning approach

To perform the prognostic model analytic classification tasks we compare five different machine learning algorithms then based on the result of the accuracy.

	Accuracy
k-Nearest Neighbors	0.914563
Support Vector Machine	0.924272
Random Forest	0.924272
Logistic Regression	0.924272
Multilayer Perceptron	0.924272

Figure 6-1 Models with their respective accuracy

The above table shows performance machine learning algorithms SVM, MLP, KNN, RF and LR. These five models are known for being robust and capable of achieving good prediction result even with low correlation, and low missing features. To increase the accuracy of the five models we used hyperparameter tuned by sklearn cross validation and their performance was evaluated on four different metrics: accuracy, precision and F1-score,

6.2. Features selection

The influence of all the features in the data are calculated by the experiment conducted. The features that show a major change in the prediction are tabulated in Table 6-1. When features with no affect in the prediction are removed, there was difference in the accuracy of prediction.

Table 6-1 Feature importance

No	Feature Name	Feature Value
1	High Fever	0.157779
2	Diabetic	0.129601
3	Blood Pressure	0.084929
4	Smoker	0.093258
5	Sex	0.092504

6.2.1. Parameter Tuning

Selecting best possible parameters for all models: Random Forest (RF), Multilayer Perceptron (MLP), Logistic Regression (LR) and SVM. Improve a model's performance by tuning its parameters. Tuning model results are listed below:

Table 6-2 Performance Metrics and Results

Models	Accuracy	Precision	Recall	F1-score
SVM	96.99	0.78	0.42	0.575
LR	96.69	0.5	0.111	0.182
RF	97.87	0.8	0.44	0.51
MLP	95.959	0.375	0.333	0.353
KNN	96.69	0.56	0.469	0.45

For COVID-19 prognosis we considered demography, clinical and laboratory findings from 1716 patients. We used five different types of models to learn and predict the findings. Later, the prognosis was performed and the performance of the machine learning applications models was evaluated. Table 6-2 shows the results of all machine learning application models. In terms of predictive performance, we observed that the overall best-identified models by performance metrics score were 97.87% by RF for prognostic COVID-19 disease. Nevertheless, the best clinical

prediction results achieved a respectable accuracy of 97.87%, f1-score of 0.51%, and recall of 0.44%, respectively with RF. From the results, we can see that RF is a good prediction model. Precision can be defined as the ratio of correctly predicted positive observations to the total predicted positive observations. In information retrieval studies, a perfect precision should be 1. In this research, the best precision score was obtained with an RF of 0.8690. Recall is the ratio of correctly predicted positive observations to all observations. Like precision, a recall score must reach to 1 for the perfect classification process. The best recall value was obtained from RF machine learning application model with 0.44. F1 score is the weighted average of precision and recall values. This evaluation criterion takes both false positives and false negatives. Getting a good F1 score indicates less false positives and low false negatives. The perfect F1 score is when the value is 1. We have the best F1 value with RF 0.51. In this study, recall is an important evaluation criterion since it is computed by taking the ratio of correctly identified COVID-19 prognostic to the total number of COVID-19 diseased patients.

6.3. Interpretability of models

The overall goal of this approach would be to provide a system to identify high-risk individuals at an early stage to help in allocating adequate resources, such as ICU beds and to provide necessary interventions before irreversible clinical damage occurs. In this model, the most important features used for prognostic were identified using feature importance. Next using SHAP explainer we try to explain the risk of Covid-19 by takes some features as input and produces some predictions as output. The choice to apply SHAP, an edge method that explains predictions made by black-box machine learning models. SHAP values to try and understand the model output on specific individuals using force plots.

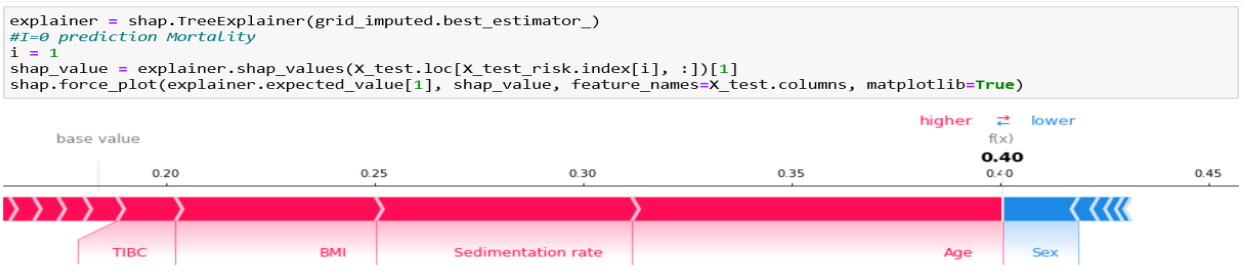

Figure 6-2 SHAP explainer

Higher Age increases the prognostic risk of cvoid-19 (i.e. the red sections on the left are features which push the model towards the final prediction in the positive direction).

The blue sections on the right are features that push the model towards the final prediction in the negative direction (if an increase in a feature leads to a lower risk, it will be shown in blue).

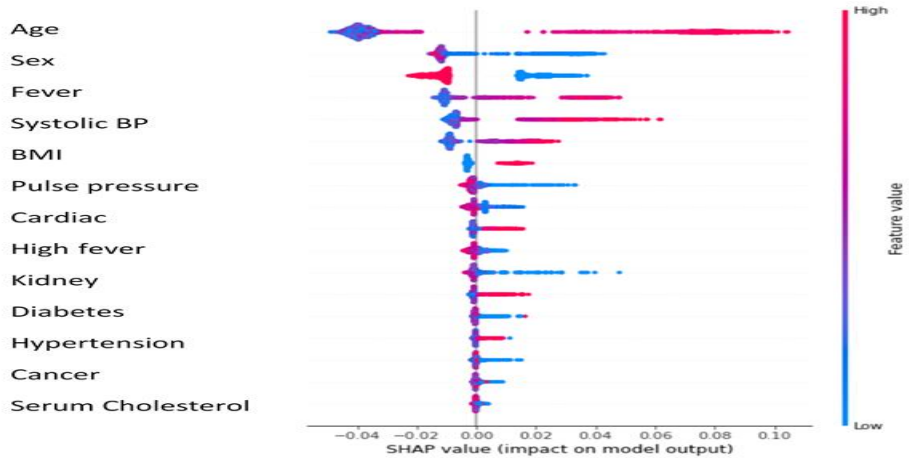

Figure 6-3 Feature Importance in SHAP

Clearly we see that being a men sex, as opposed to women for which has a negative SHAP value, meaning that it increase the risk of dying on covid-19. High age and high systolic blood pressure have positive SHAP values, and are therefore related to increased mortality.

It can be seen how features interact using dependence plots. These plot the SHAP value for a given feature for each data point, and color the points in using the value for another feature.

```
shap.dependence_plot('Age', shap_values, X_test, interaction_index='Sex')
```

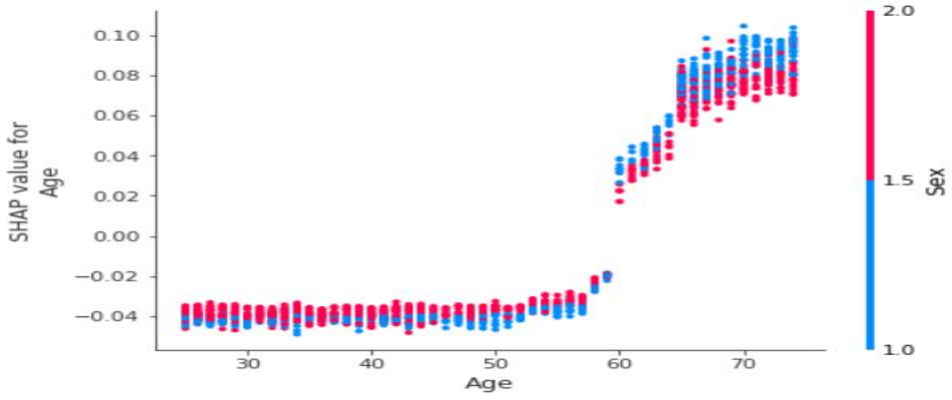

Figure 6-4 Interaction between age and sex

We see that while Age > 50 generally shows (positive SHAP value), being men and the risk impact of age.

6.4. Discussion

This chapter provides a detailed analysis of the results of the experiments performed in the previous chapter. In essence, experiments are carried out to build five separate supervised machine learning models. The performance of all the models will be evaluated based on the accuracies obtained after training each model on the pre-processed dataset. After presenting the main algorithms and analyzing the data pre-processing and cross-validated resampling techniques in theory, five typical machine learning algorithms (Logistic Regression, Sector Vector Machine, K-Nearest Neighbor Multiline Perceptron and Random Forest) are implemented on a real dataset, and the corresponding performance of the built models are quantitatively and visually evaluated in details.

The model developed with Random Forest happened to be the best model among all models developed in terms of accuracy with 97.87 % when compared with other models developed with LR, KNN, SVM and MLP which have 95.65%, 96.6%, 96.33% and 95% accuracy respectively.

Random forest model indicated that the age feature is the most important feature among all the dependent features of the dataset including the clinical features. The model indicates that most of the people with older age are impact to be death in SARS-CoV-2 when compared to people with lower ages. Regarding gender, males are more prone to COVID-19 mortality than females, and those who smoke tobacco are more likely to be dead with than non-tobacco smokers.

The model will help the health workers with the prognosis of the COVID-19 outcome, and this will reduce the huge burden on healthcare systems. The supervised ML models can be used as retrospective evaluation techniques or tools to validate COVID19 outcomes. This study shows how ML prognostic COVID-19 outcome models can be developed, validated and used as the tools for rapid prognostic of COVID-19 outcome. The study also shows the important roles playing by supervised ML algorithms in the prognostic of the COVID-19 pandemic, which can help reduce the huge burden on limited healthcare systems.

CHAPTER SEVEN

7. CONCLUSIONS AND FUTURE WORKS

7.1. Conclusions

This research proposed to build a prognostic model for COVID-19 using machine learning approaches. This study attempted to implement and compare machine learning methods specifically for COVID-19 prognostic. To successfully execute the study, it was essential to understand and define COVID-19 prognostic factors, explore existing various techniques used to tackle the problem, understand the COVID-19 outcome, as discussed in chapter two. Also, the different methods followed to implement and design models that have the capability of prognostic the outcome.

This paper measure the performance of five prognostic models built on SVM, RF, KNN, LR and MLP methods. These models are used to prognostic COVID-19 using various parameters provided in the COVID-19 dataset. 1716 data samples are collected. In the first stage of the study, the data were standardized and then used as inputs for the machine learning models then classification was carried out and the performances of the models were measured with precision, recall, accuracy, and F1- scores. To validate the models, we applied 10 fold cross-validation approaches. In 10 fold cross-validation strategy, the best meaningful results were observed from RF machine learning model with an f1-score of 0.44%, precision 0.8%and recall of 0.55%, the selected machine learning models developed in the study showed an accuracy of over 97.87. Similar inferences can be made for precision and recall values. In conclusion, we found evidence to suggest that machine learning application models can be applied to prognostic COVID-19 infection with prognostic factors. Our experimental results indicate that may be useful to help prioritize scarce healthcare resources by assigning personalized risk scores using laboratory, demography, and clinical analysis data. In addition to these, our findings on the importance of laboratory measurements towards predicting COVID-19 infection for patients increase our understanding of the outcomes of COVID- 19 disease. Based on our study’s results, we conclude that health- care systems should explore the use of prognostic models that assess individual COVID-19 risk to improve healthcare resource prioritization and inform patient care.

7.2. Recommendations

This study used machine learning approaches to build a model for COVID-19 prognostic outcome. As a result, the researcher recommends COVID-19 treatment centers to develop future deep learning prognostic models that assist the healthcare system to forecast the outcome of the patient.

7.3. Future Works

In the future, we can forecast Covid-19 cases for different countries with comparative analysis. As the Covid19 cases are increasing exponentially it is impossible to defeat this pandemic without the inception of Artificial Intelligence that can help in proper treatment, prevention and vaccine development. Therefore, we can compare the leading technologies and vaccines used or developed by various countries to drawback Covid-19 impacts and enhance its future timeline.

.

8. REFERENCES

[1] G. C. Study, ""The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2"," Nature Microbiology, April(2020), p. 536–544.

[2] C. f. D. C. a. Prevention, "Symptoms of Coronavirus," February(2021).

[3] CDC, ""Underlying Medical Conditions Associated with High Risk for Severe COVID-19: Information for Healthcare Providers"," CDC, May 13, 2021.

[4] "worldometer," 13 August 2021. [Online]. Available: https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1?%20.

[5] "Covid," April 2020. [Online]. Available: https://graphics.reuters.com/world-coronavirus-tracker-and-maps/countries-and-territories/ethiopia/.

[6] WHO, "Ethiopia," 13 August 2021. [Online]. Available: https://covid19.who.int/region/afro/country/et.

[7] T. Girum, ""Global strategies and effectiveness for COVID-19 prevention through contact tracing, screening, quarantine, and isolation"," Tropical Medicine and Health, November 23, 2020.

[8] H. AD, "Prognosis research strategy (PROGRESS) 4: stratified medicine research.," BMJ, 2013.

[9] M. KG, ""Prognosis and prognostic research: what, why, and how?"," BMJ, pp. 1-9, 2009.

[10] W. JH, Clinical prediction rules applications and methodological standards, N Engl J Med, 1985.

[11] D. Riley, "Prognosis Research in Health Care: Concepts, Methods, and Impact", Oxford University PressPrint Publication, Jan 2019.

[12] Z. Z, "Prediction model and risk scores of icu admission and mortality in covid-19," PloS One, p. 15, 2020.

[13] S. H. HUANG, ""Prognostic factors for covid-19 pneumonia progression to severe symptoms based on earlier clinical features: a retrospective analysis"," Google Scholar, pp. 1-17, 2020.

[14] Y. H, "Severity detection for the coronavirus disease 2019 (covid-19) patients using a machine learning model based on the blood and urine tests.," Frontiers, 2020.

[15] S. S. 2. Helen Barratt 2009, "Studies of disease prognosis," 2018. [Online]. Available: https://www.healthknowledge.org.uk/public-health-textbook/research-methods/1a-epidemiology/sudies-disease-prognosis.

[16] W. G. R. S. e. a. Laupacis A, "How to use an article about prognosis.," JAMA , p. 272:234–237, 1994;.

[17] e. a. Kent DM, Limitations of applying summary results of clinical trials to individual patients: the need for risk stratification, JAMA, 2007.

[18] C. GS, "Identifying patients with undetected renal tract cancer in primary care: an independent and external validation of QCance", Cancer Epidemiol., 2013.

[19] e. a. Harrell FE Jr, "Multivariable prognostic models:issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors", Stat Med, 1996.

[20] H. N. A. B. P. Vera Ehrenstein, "Clinical epidemiology in the era of big data: new opportunities, familiar challenges," PMC, 2017 Apr 27..

[21] J. S. Ash, ""Some unintended consequences of information technology in health care: the nature of patient care information system-related errors"," PubMed, Nov 21, 2003.

[22] M. E. Mark Esposito, "What is machine learning?," 3 May 2017. [Online]. Available: https://theconversation.com/what-is-machine-learning-76759.

[23] Ganegoda, ""Involvement of Machine Learning Tools in Healthcare Decision Making"," Hindawi, 27 Jan 2021.

[24] A. S. Ahuja, ""The impact of artificial intelligence in medicine on the future role of the physician"," PMC Labs, pp. 5-10, 2019.

[25] T. O. Ayodele, ""Types of machine learning algorithms".," Publisher: InTech, p. 19–48, 2010..

[26] J. Tolles and W. J. Meurer, "Logistic Regression Relating Patient Characteristics to Outcomes," Jama, (2016).

[27] P. Marius-Constantin, "“Multilayer perceptron and neural networks"," WSEAS Trans. Circuits Syst., vol. 8, no. 7,, p. . pp 579– 588, 2009.

[28] Y. Qi, Random forest for bioinformatics. In Ensemble machine learning,, Springer, 2012. , pages 307–323.

[29] I. Yilmaz, Comparison of landslide susceptibility mapping methodologies for Koyulhisar, Turkey: conditional probability, logistic regression, artificial neural networks, and support vector machine., Turkey: Springer, 2010.

[30] Augusta, ""Heart Failure: Risk Factors"," 2021. [Online]. Available: https://www.universityhealth.org/heart-failure/risk-factors/.

[31] Z. X. Zhang, "Prognostic factors for mortality due to pneumonia among adults from different age groups in Singapore and mortality predictions based on PSI and CURB-65," PMC, 2018.

[32] Pinheiro, "Mortality Predictors and Associated Factors in Patients in the Intensive Care Unit: A Cross-Sectional Study," Hindawi, pp. 1-10, 2020.

[33] A. E. M. Booth AL, ""P. Development of a prognostic model for mortality in covid-19 infection using machine learning"," Mod Patho, pp. 1-10, 2020.

[34] S. Fs, "Comorbidities and Risk Factors for Severe Outcomes in COVID-19 Patients in Saudi Arabia: A Retrospective Cohort Study," Dovepress, 2021.

[35] Han, ""Lactate dehydrogenase, a risk factor of severe covid-19"," medRxiv, 2020.

[36] A. Boothet.al, ""Development of a prognostic model for mortality in covid-19 infection using machine learning"," Google Scholar, 2020.

[37] S. S, ""A prediction model to prioritize individuals for sars-cov-2 test built from national symptom surveys"," Med, pp. 196-208, 2020.

[38] e. a. Izquierdo JL, "Clinical characteristics and prognostic factors for intensive care unit admission of patients with covid-19:," J Med Internet, 2020.

[39] T. T, "Machine learning prediction of sars-cov-2 polymerase chain reaction results with routine blood tests," Lab Med, 2020.

[40] C. François, "Deep Learning with Python," New York: Manning Publications, 2017.

[41] C. Francois, "Deep Learning with python," New York: Manning publication, 2017.

[42] e. a. Lars Buitinck, "API design for machine learning software: Experiences from the scikit-learn project," Research Gate, 2013.

[43] R. a. Z. B. Bellazzi, ""Predictive data mining in clinical medicine"," biolab, pp. 1-17, 2008.

[44] B. S. THOMAS, "Data Cleaning in Machine Learning: Best Practices and Methods," 11 december 2019. [Online]. Available: https://www.einfochips.com/blog/data-cleaning-in-machine-learning-best-practices-and-methods/.

[45] e. a. Sterne JA, ""Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls"," BMJ, 2009.

[46] sergiosantoyo, "A Brief Overview of Outlier Detection Techniques," 11 Sep 2017. [Online]. Available: https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561?gi=5b4a1c2f3a8.

[47] C. Liu, "Data Transformation: Standardization vs Normalization," August 2019. [Online]. Available: https://www.kdnuggets.com/2020/04/data-transformation-standardization-normalization.html.

[48] R. Walimbe, "Handling imbalanced dataset in super," August 2017. [Online]. Available: www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family.

[49] P. Płoński, "Random Forest Feature Importance Computed in 3 Ways with Python," 29 June 2020. [Online]. Available: https://mljar.com/blog/feature-importance-in-random-forest/.

[50] "Converting YOLO* Models to the Intermediate Representation (IR)," [Online]. Available: https://docs.openvinotoolkit.org/latest/_docs_MO_DG_prepare_model_convert_model_tf_specific_Convert_YOLO_From_Tensorflow.html.

[51] P. a. G.Adam, "Deep learning a racttioner's Approch, Sebastopol:," O'Reilly Media, 2017.

[52] J. Brownlee, "Ordinal and One-Hot Encodings for Categorical Data," 12 june 2020. [Online]. Available: https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/.

[53] S. K. a. Zisserman, "A Very Deep Convolutional Networks for LargeScale Image Recognition," arXiv preprint arXiv, vol. 1409.1556, 2014.

[54] James, "Machine Learning Crash Course Google Developers," Google, 2019. [Online]. Available: https://developers.google.com/machinelearning/crash-course/classification/true-falsepositive-negative..

[55] J. Brownlee, ""A Gentle Introduction to Ensemble Learning Algorithms"," 27 April 2021. [Online]. Available: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/.

[56] W. L, "Prediction models for diagnosis and prognosis of Covid19 infection: Systematic review and critical appraisal," BMJ-Brit Med, June 2,2020.

9. APPENDIX

9.1. A.1 Sample Source Code to Model KNN, SVM, RF and MLP

```
# Initial tool settings

%matplotlib inline

warnings.filterwarnings('ignore')

warnings.filterwarnings('ignore', category=DeprecationWarning)

plt.style.use('fivethirtyeight')

sns.set(style='white', context='notebook', palette='deep')

py.init_notebook_mode(connected=True)

random_state = 43

names = ["k-Nearest Neighbors",

        "Support Vector Machine",

        "Random Forest",

        "Logistic Regression",

        "Multilayer Perceptron",

       ]

classifiers = {  "k-Nearest Neighbors" : KNeighborsClassifier(n_neighbors=3),

                 "Support Vector Machine" :  SVC(random_state=random_state),

                 "Random Forest" : RandomForestClassifier(random_state=random_state),

                 "Logistic Regression" : LogisticRegression(random_state=random_state),

                 "Multilayer Perceptron" :
MLPClassifier(hidden_layer_sizes=(100,),momentum=0.9,solver='sgd',random_state=random_st
ate),
```

```
            }
algorithms = [ KNeighborsClassifier(n_neighbors=3),
            SVC(random_state=random_state),
            RandomForestClassifier(random_state=random_state),
            LogisticRegression(random_state=random_state),

MLPClassifier(hidden_layer_sizes=(100,),momentum=0.9,solver='sgd',random_state=random_st
ate),
            ]
```

9.2. A.2 SVM tunning

```
svc = SVC(probability=True, random_state=random_state)
# Set the parameters by cross-validation
svc_pg = [{'kernel': ['rbf'], 'gamma': [1e-1, 1e-2, 1e-3, 1e-4],'C': [1, 10, 100, 1000]},
        {'kernel': ['linear'], 'C': [1, 10, 100, 1000]}]
scores = ['precision', 'recall']
for score in scores:
    svc.fit(X_train, y_train)
    y_eval = svc.predict(X_test)
    acc = sum(y_eval == y_test) / float(len(y_test))
    print("Accuracy of SVC: %.2f%%" % (100*acc))

    print("Tuning hyper-parameters for %s\n" % score)
    svc_gscv = GridSearchCV(svc, svc_pg, cv=kfold, scoring='%s_macro' % score)
```

```
svc_gscv.fit(X_train, y_train)
print("Best parameters set found on training set:")
print(svc_gscv.best_params_,"\n")
print("Grid scores on training set:")
means = svc_gscv.cv_results_['mean_test_score']
stds = svc_gscv.cv_results_['std_test_score']
    for mean, std, params in zip(means, stds, svc_gscv.cv_results_['params']):
    print("%0.3f (+/-%0.03f) for %r" % (mean, std * 2, params))
print("Detailed classification report:")
print("The model is trained on the full training set.")
print("The scores are computed on the full test set.")
y_true, y_pred = y_test, svc_gscv.predict(X_test)
print(classification_report(y_true, y_pred))
svc_est = svc_gscv.best_estimator_
svc_score = svc_gscv.best_score_
print("Best estimator for parameter C: %f\n" % (svc_est.C))
print("Best score: %0.2f%%\n" % (100*svc_score))
#print(clf)
```

9.3. A.3 MLP Tunning

```
mlp = MLPClassifier(momentum=0.15,solver='sgd',learning_rate_init=1.0, early_stopping=True, shuffle=True,random_state=random_state)


mlp_pg={
```

'learning_rate': ["constant", "invscaling", "adaptive"],

'hidden_layer_sizes': [(10,10),(100,),(100,10)],

#'hidden_layer_sizes': [x for x in itertools.product((10,20,30,40,50,100),repeat=3)],

#'tol': [1e-2, 1e-3, 1e-4],

#'epsilon': [1e-3, 1e-7, 1e-8],

'alpha': [1e-2, 1e-3, 1e-4],

#'activation': ["logistic", "relu", "Tanh"]

}

mlp_gscv = GridSearchCV(mlp,param_grid=mlp_pg, cv=kfold, scoring="accuracy", n_jobs= -1, verbose = 1)

mlp_gscv.fit(X_train, y_train)

mlp_est = mlp_gscv.best_estimator_

mlp_score = mlp_gscv.best_score_

print("Best estimator:", mlp_est, "\nBest Score:", mlp_score)

9.4. A.3 LR Tuning

```
lr = LogisticRegression(
    C=0.1,
    penalty='l2',
    dual=True,
    tol=0.0001,
    fit_intercept=True,
    intercept_scaling=1.0,
    class_weight=None,
    random_state=random_state)
```

```
lr_pg = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}


lr_gscv = GridSearchCV(lr,param_grid=lr_pg, cv=kfold, scoring="accuracy", n_jobs= -1, verbose = 1)

lr_gscv.fit(X_train, y_train)

lr_est = lr_gscv.best_estimator_

lr_score = lr_gscv.best_score_

print("Best estimator:", lr_est, "\nBest Score:", lr_score)
```

9.5. A.4 KNN Tuning

```
k_range=list(range(1,50))

data_copy=pd.Series()

k_scores = []

#x=[0,1,2,3,4]

for i in k_range:
    model=KNeighborsClassifier(n_neighbors=i)
    scores = cross_val_score(model, X, y, cv=kfold, scoring='accuracy')
    k_scores.append(scores.mean())
    model.fit(X_train,y_train)
    pred_test=model.predict(X_test)
    data_copy=data_copy.append(pd.Series(metrics.accuracy_score(pred_test,y_test)))


#plt.plot(k_range, k_scores)

plt.plot(k_range, data_copy)

plt.xlabel('Value of K for KNN')

plt.ylabel('Cross-Validated Accuracy')

#plt.xticks(X)

plt.show()

print('Accuracies for different values of n are:\n', data_copy.values)
```