EARLY PREDICTION OF COVID-19 USING MACHINE LEARNING TECHNIQUES

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

GURRAMPATI USHASWI	(21UECM0092)	**(VTU 20205)**
PAILA LOKESH KUMAR	(21UECM0175)	**(VTU 20215)**
MADEM HARSHITHA	(21UECM0292)	**(VTU 19287)**

Under the guidance of
Dr.K.ANTONY KUMAR,M.E., Ph.D.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

January, 2024

EARLY PREDICTION OF COVID-19 USING MACHINE LEARNING TECHNIQUES

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

GURRAMPATI USHASWI	(21UECM0092)	**(VTU 20205)**
PAILA LOKESH KUMAR	(21UECM0175)	**(VTU 20215)**
MADEM HARSHITHA	(21UECM0292)	**(VTU 19287)**

Under the guidance of
Dr.K.ANTONY KUMAR,M.E., Ph.D.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

January, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”EARLY PREDICTION OF COVID-19 USING MACHINE LEARNING TECHNIQUES ” by ”GURRAMPATI USHASWI (21UECM0092), PAILA LOKESH KUMAR (21UECM0175), MADEM HARSHITHA (21UECM0292)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Dr.K.Antony Kumar
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January, 2024

Signature of Head of the Department
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January, 2024

Signature of the Dean
Dr. V. Srinivasa Rao
Professor & Dean
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

GURRAMPATI USHASWI
Date: / /

PAILA LOKESH KUMAR
Date: / /

MADEM HARSHITHA
Date: / /

APPROVAL SHEET

This project report entitled EARLY PREDICTION OF COVID-19 USING MACHINE LEARNING TECHNIQUES by GURRAMPATI USHASWI (21UECM0092), PAILA LOKESH KUMAR (21UECM0175), MADEM HARSHITHA (21UECM0292) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr.K ANTONY KUMAR,M.E., Ph.D.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr. M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Dr. K. ANTONY KUMAR,M.E., Ph.D.,** for his cordial support, valuable information and guidance, he helped us in com-pleting this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E., Mr. SHARAD SHANDHI RAVI, M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

GURRAMPATI USHASWI (21UECM0092)
PAILA LOKESH KUMAR (21UECM0175)
MADEM HARSHITHA (21UECM0292)

ABSTRACT

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus.A family of viruses known as coronaviruses is responsible for diseases like the common cold, Middle East respiratory syndrome (MERS), and severe acute respiratory syndrome (SARS).Everyone is currently dealing with the COVID-19 issue, so we need to anticipate the virus in order to take some security steps. Having early warning systems in place to predict the disease is critical. The number of corona virus confirmed cases and deaths worldwide can now be estimated using predictive algorithms. It contains techniques for projecting future cases using current data.Support vector regression modeling is used in the proposed work to forecast the total number of deaths, recovered cases, cumulative number of confirmed cases, and number of cases each day. Using a support vector regression model with the Radial Basis Function as the kernel and a 10 percent confidence interval for curve fitting, the suggested methodology predicts values. The regression score, mean square error, root mean square error, and percentage accuracy are the metrics used to derive the model performance parameters. The model predicts deaths, recoveries, the total number of confirmed cases with greater than 97 percent accuracy, and daily new cases with an accuracy of 87 percent. The data point to a Gaussian decline in the number of cases, and it might take a further three to four months for the minimal level to be reached without any new cases being reported. The approach is more accurate and very efficient than polynomial or linear regression.

Keywords: Linear Regression, Machine Learning, Polynomial Regression, Radial Basis Function kernel, Support Vector Machine.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

CNN	Convolutional Neural Network
COVID	Coronavirus Disease
LR	Linear Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
PR	Polynomial Regression
RBF	Radial Basis Function
SVM	Support Vector Machine
SVR	Support Vector Regression

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Coronavirus Disease-19(COVID-19), caused an epidemic in Wuhan, China, in December 2019. It quickly expanded to over 200 nations throughout the world following its inception. The number of verified corona virus infected patients is expanding at an exponential rate throughout the world. It has infected over 17 million individuals to far, with over 0.7 million people are dying as a result. It's critical to develop a technological tool that can estimate the number of infected individuals and forecast the worst-case situations.The tests that measure the Coronavirus to determine the presence of illness are to be examined more since the range of symptoms of positive cases has been rising. As corona virus has achieved pandemic status and the number of patients continues to rise at an exponential pace is critical in determining and controlling the spread of this rapidly spreading illness. The main goal is to create a model that predicts the number of confirmed,deaths and recovery cases caused by Corona virus through out the world based on data from several countries.This can predict Corona cases for next 10 days.

One of the fields of application of Machine Learning(ML) is the area of health. ML has led different researchers to approach from this technique of corona virus. It is a contagious virus that belongs to the coronavirus family. The disease causes flu like symptoms such as cough, fever, fatigue, and shortness of breath. The primary source of the virus is still under debate. It has spread worldwide, generating difficulties in different aspects of human life.The virus has had variants such as delta and omicron that have caused different waves (high peaks) of infections and deaths worldwide. The Omicron variant, considered more transmissible but less lethal, was shown to have reached 61.5 percent of women that reported infections. The corona virus has been spread more than 491 million confirmed cases, and more than 6.1 million deaths. The scientific community is developing vaccines, techniques, and procedures using different technologies that using tools and is investigating problems to improve the characterization of ML algorithms with better performance to perform survival analysis studies.

Support Vector Machines (SVM) have emerged as powerful tools in the realm of artificial intelligence for making early predictions about the COVID-19 disease. SVM is a supervised learning algorithm that proves valuable in classification and regression tasks. In the context of COVID-19 early predictions, SVM utilizes historical data, considering various factors such as patient demographics, regional statistics, and healthcare infrastructure. By leveraging machine learning techniques, SVM models can identify subtle patterns and relationships within the data that may not be immediately apparent through traditional analysis. This allows for the creation of predictive models capable of forecasting potential outbreaks, aiding in resource allocation, and informing public health measures.

The integration of artificial intelligence with SVM in early predictions of COVID-19 brings about significant advancements in healthcare analytics. SVM models, when applied to COVID-19 datasets, can learn complex patterns from a multitude of variables, providing a more nuanced understanding of the disease's spread. Moreover, SVM excels in handling high-dimensional data, allowing for the inclusion of diverse parameters influencing the trajectory of the pandemic. As a result, these AI-driven SVM models contribute to more accurate and timely predictions, aiding authorities in proactive decision-making and risk mitigation strategies. By harnessing the power of SVM and artificial intelligence, researchers and healthcare professionals can gain valuable insights into the early stages of the COVID-19 disease, ultimately fostering more effective and data-driven responses.

1.2 Aim of the project

The primary goal of the experiment is to predict the virus for the next ten days. We need a strong model that predicts how the virus could spread across different countries and regions. The overall number of instances, as well as active and fatal cases, are detected by our project. Identifying the most suitable machine learning technique for prediction, to perform on clinical reports of patients.

1.3 Project Domain

In project domains, the utilization of past data to train machine learning models forms a fundamental step in constructing predictive systems across various industries. This process involves exposing the model to historical datasets, enabling it to discern intricate patterns and relationships within the data. Subsequently, the trained model undergoes rigorous testing, employing a distinct subset of past data allocated specifically for evaluation. This testing phase is imperative to gauge the model's ability to generalize to novel, unseen data, ensuring its robustness in real-world applications. Key performance metrics, such as accuracy, are then employed to quantitatively measure the effectiveness of the trained machine learning model in accurately predicting outcomes.

Upon successful training and testing, the machine learning model transitions to the prediction phase, where it is deployed to make informed predictions on new, incoming data. Continuous monitoring and refinement are essential to adapt the model to evolving patterns in the data distribution. Evaluation of the model's performance remains an ongoing endeavor, involving the use of artificial intelligence techniques to enhance adaptability. As the model operates in real-world scenarios, its reliability and accuracy are scrutinized using subsets of available past data, ensuring that it continues to meet the specified performance measures. This iterative cycle, coupled with the integration of AI, establishes a dynamic and efficient system for leveraging machine learning in making predictions based on historical information.

1.4 Scope of the Project

In this project, a robust time series model has been meticulously developed through machine learning analysis to forecast the progression of the coronavirus. The primary objective of our research is to assess the efficacy of these models in predicting COVID-19 outcomes in patients by identifying the most suitable machine learning technique for prediction. Leveraging advanced machine learning techniques, the model is designed to analyze temporal patterns in the data, offering insights into the potential trajectory of the virus. This predictive capability holds significant implications for proactive healthcare interventions and resource allocation. The focus is on refining the accuracy and reliability of the time series model to enhance its practical utility in forecasting COVID-19 cases, thereby contributing to the ongoing global efforts in combating the pandemic.

In conjunction with the development of the time series model, our research endeavors extend to the identification of key features within the clinical information of patients that exert influence on the predictive outcomes. By delving into the rich dataset encompassing clinical details, we aim to unravel the nuanced relationships between various features and the progression of COVID-19. This comprehensive analysis not only enhances the predictive accuracy of the machine learning model but also sheds light on critical factors that can significantly impact patient outcomes. The integration of artificial intelligence in this research ensures a sophisticated approach to feature identification and model refinement, fostering a more nuanced understanding of COVID-19 prediction based on clinical parameters.

Chapter 2

LITERATURE REVIEW

Alazab et al. (2020) pioneered a groundbreaking study employing deep learning, specifically a Convolutional Neural Network (CNN), for COVID-19 detection through chest X-ray images. Demonstrating an impressive F-measure range of 95–99 percent, the system exhibits promising performance in predicting confirmations, recoveries, and deaths. Identifying coastal areas as significantly impacted, the research achieves notable accuracy, 94.80 percentage in Australia and 88.43 percentage in Jordan. This innovative approach holds the potential to reshape global pandemic efforts, providing authorities and healthcare professionals with valuable insights. Alazab et al.'s research marks a pivotal advancement in leveraging artificial intelligence for effective COVID-19 detection and mitigation strategies.

Arora et al. (2020) conducted a significant study employing deep learning, featuring Convolutional Long Short-Term Memory(LSTM) and Bidirectional LSTM models, to predict COVID-19 positive cases across Indian states and union territories. Utilizing data from the Ministry of Health and Family Welfare, the study categorizes regions by severity, aiding in short-term case number predictions. The models prove effective for state authorities, facilitating medical infrastructure management and informed decisions on lockdowns and economic activities. Comparative analysis underscores the LSTM model's effectiveness. This research stands as a crucial contribution, offering valuable insights for public health interventions and evidence-based policy-making in the context of the COVID-19 pandemic.

Parbat et al.(2020) contributed to COVID-19 literature with a Python-based support vector regression model for predicting cases in India. Utilizing data from March 1st to April 30th, 2020, the study showcases the model's remarkable performance, boasting over 97 percentage of accuracy in predicting deaths, recoveries, and cumulative cases. The methodology encompasses data preparation, visualization, and prediction, emphasizing the decreasing Gaussian trend in the time series. The model's robustness in handling dataset variability is underscored, offering potential for informed policy-making and crisis management. This research serves as a valuable resource, showcasing the efficacy of advanced modeling techniques in the context of pandemic prediction and control.

Rath et al.(2020) contributed to COVID-19 literature by employing regression models for trend prediction. Their study, focusing on India and Odisha, utilizes multiple linear regression to forecast active cases. In a growing body of research utilizing regression for pandemic analysis, their work stands out for emphasizing advanced techniques. The findings underscore the model's strong predictive ability, urging the implementation of enhanced measures to curb the virus. Despite its contributions, the study acknowledges limitations, addressing external factors like particulate matter's influence on outcomes.

Singh et al. (2020) contributed a significant review to COVID-19 prediction research through their study utilizing SVM. Addressing limitations of traditional approaches, the research spans from January 22, 2020, to April 25, 2020, globally. Employing SVM, the study forecasts confirmed cases, deaths, and recoveries, emphasizing the efficacy of advanced modeling for outbreak severity estimation. The findings underscore the pivotal role of SVM in pandemic forecasting, offering valuable insights crucial for informed public health and policy decisions in managing and controlling the ongoing COVID-19 crisis. This research is a pivotal addition to the discourse on employing machine learning for pandemic prediction and mitigation.

Ahmad et al.(2021) contributed to COVID-19 research by employing machine learning, specifically a shallow single-layer perceptron neural network and Gaussian process regression, for predictive modeling. Focused on aiding public health management, their study aims to enhance strategies for handling the pandemic. While the findings offer potential improvements for COVID-19 public health management, the implications of machine learning in informing policy and decision-making warrant careful consideration. The authors, declaring no conflicts of interest and receiving no specific funding, underscore the importance of thoughtful application and evaluation of machine learning techniques in the context of public health response to the ongoing pandemic.

Guhathakurata et al.(2021) introduced a pioneering approach to COVID-19 prediction, employing SVM. The study emphasizes the critical importance of accurate prediction, addressing limitations in existing methods. The novel SVM model exhibits an impressive 87 percentage accuracy in forecasting severe infections, underscoring its significance in combatting the pandemic. Proposing the use of visual programming with the Orange toolkit, the paper not only provides a comparative analysis of machine-learning models but also advocates for a comprehensive approach to COVID-19 prediction. The research stands as a valuable contribution, shedding light on the potential of SVM to navigate the challenges inherent in understanding and forecasting the virus.

Khodeir et al.(2021) presented a crucial meta-analysis on early prediction keys for COVID-19 progression. This collaborative effort by experts from diverse institutions yields valuable insights for healthcare professionals, researchers, and policymakers. The study identifies inflammatory markers, cortisol levels, abnormal coagulation, and multiorgan injury as key predictors, emphasizing their potential in guiding early interventions and improving case management. By amalgamating existing knowledge, this research significantly contributes to our understanding of COVID-19 dynamics. The findings hold practical implications for real-world applications, enabling enhanced strategies for public health management and policy formulation amid the ongoing global pandemic.

Tiwari et al.(2022) contributed a significant review, delving into the landscape of ML algorithms for COVID-19 case analysis. The study explores the nuanced impact of data type and nature, emphasizing the pivotal role of intelligent approaches, particularly ML, in navigating the pandemic. Notably, the review encompasses the evolution of ML algorithms and provides a curated list of relevant studies and datasets. Offering insights into future research directions and community resources, this comprehensive review stands as a valuable asset for data scientists, researchers, and policymakers aiming to harness ML in the ongoing battle against the Covid-19 epidemic.

Gothai et al.(2023) contributed to COVID-19 research with a focus on growth prediction using machine learning. Leveraging a dataset from John Hopkins University, the study employs linear regression, support vector machine, and Holt-Winters model for trend forecasting. Notably, the research identifies Holt-Winters Exponential Smoothing as the most accurate method for predicting global confirmed cases. Despite these advancements, the study conscientiously acknowledges potential biases and limitations inherent in machine learning models predicting the intricate spread of COVID-19. By blending diverse algorithms, the research provides valuable insights, emphasizing the nuanced nature of predictive modeling in the context of a complex and evolving pandemic.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The global spread of COVID-19 has ignited a widespread discourse regarding the efficacy of the existing healthcare models, particularly patient-centered and hospital-centered approaches, known for their high costs. The spread of COVID-19 has spurred a compelling need to explore more economically viable alternatives while maintaining the quality of healthcare services. The escalating demand for healthcare resources necessitates a more efficient allocation, prompting discussions within the medical community on how to achieve this balance without compromising patient care.

In the context of global spread, there is a growing interest in investigating remotely managed healthcare delivery systems as a potential solution. The approach may be particularly applicable in cases involving patients with mild illnesses and no significant risk factors. The medical fraternity is actively engaged in weighing the advantages and disadvantages of such a system, contemplating its feasibility, benefits, and potential challenges to ensure a well-informed and responsible transition towards more sustainable and accessible healthcare practices.

Disadvantages:

- Remote healthcare may lack the ability to conduct comprehensive physical examinations, limiting the accuracy of diagnoses and potentially leading to oversight in patient care.
- Transmitting sensitive medical information electronically poses cybersecurity risks, requiring robust measures to safeguard patient confidentiality and comply with data protection regulations.
- The absence of face-to-face communication may lead to misunderstandings or misinterpretations of medical information, potentially affecting the accuracy of treatment plans and patient understanding.

3.2 Proposed System

The proposed system follows a systematic approach, commencing with the training of the model using a dedicated dataset. In the initial phase, the model learns and adapts to the patterns present in the data. Following the training, the subsequent steps involve data manipulation, preprocessing, and feature extraction on the incoming datasets. This crucial preprocessing phase ensures that the data is appropriately formatted and relevant features are extracted, setting the stage for accurate predictions.

In the second phase of the process, the preprocessed dataset is input into the trained model. Leveraging regression analysis models tailored for exposed Coronavirus cases, the system predicts the rate of disease spread for the upcoming 7 days. The prediction is driven by significant categorical variables identified during the feature extraction process. This entire sequence of training, preprocessing, and prediction is systematically repeated across various datasets, reinforcing the system's reliability and adaptability to diverse data scenarios. The iterative nature of the process ensures that the model generalizes well and provides robust predictions across different datasets.

Advantages:

- The systematic training process ensures that the model is well-equipped to understand and adapt to the patterns present in the data, enhancing its predictive capabilities.
- The inclusion of data manipulation and preprocessing steps enhances the quality of input data, ensuring it is appropriately formatted and contains relevant features. This contributes to the accuracy of predictions by eliminating noise and irrelevant information.
- The system's capability to predict the rate of disease spread for the upcoming 7 days provides valuable insights for proactive decision-making and resource allocation in managing the COVID-19 pandemic.

3.3 Feasibility Study

The registration of clinical studies related to COVID-19 has shown commendable speed and quality in study design. However, challenges persist in terms of feasibility, such as insufficient sample sizes, coordination issues among multidisciplinary teams, and weak quality control in the research process. The factors can potentially hinder the reliability and generalizability of study findings, emphasizing the ongoing need for addressing logistical and methodological challenges in COVID-19 research.

In parallel, the integration of machine learning, specifically using SVM algorithms, has emerged as a promising approach for predicting COVID-19 dynamics. Feasibility studies employing SVM algorithms have demonstrated potential in forecasting disease trends and identifying risk factors. The application of machine learning not only introduces a technological dimension but also addresses some of the feasibility concerns associated with traditional clinical studies. As the field evolves, the combination of clinical studies and machine learning methodologies holds promise for enhancing our understanding of COVID-19 and improving predictive capabilities.

3.3.1 Economic Feasibility

Economic feasibility is a critical aspect of project evaluation that involves a comprehensive cost-benefit analysis to assess the viability and potential returns associated with a proposed initiative. Organizations engage in this evaluative process to make informed decisions about resource allocation. Through a meticulous examination of costs and benefits, economic feasibility provides decision-makers with a clear understanding of the financial implications tied to a project. Economic feasibility involves identifying and quantifying both direct and indirect costs, including initial investments, ongoing operational expenses, and potential risks.

Moreover, economic feasibility serves as an indispensable tool for independent project assessment. By objectively scrutinizing the projected costs and benefits, organizations gain insights into the potential economic impact of the proposed project. The evaluative framework not only enhances the transparency and credibility of the project but also aids decision-makers in determining the positive economic benefits it is expected to deliver to the organization. Ultimately, economic feasibility acts as a guiding compass for strategic decision-making, helping organizations make well-informed choices that align with their financial objectives and overall sustainability.

3.3.2 Technical Feasibility

Technical feasibility is a critical evaluation encompassing the hardware, software, and technical requirements essential for implementing a proposed system. In the case of a project focused on predicting and analyzing the global spread of coronavirus, meticulous scrutiny is applied to ensure the model's compatibility with hardware infrastructure and the suitability of its software components. The assessment ensures that the technical aspects of the project are not only feasible but also well-equipped to handle the computational complexities inherent in modeling and predicting pandemic dynamics.

The model's capacity to fit and evaluate through public datasets, specifically those detailing daily confirmed active COVID-19 cases, further underscores its technical feasibility. By leveraging real-world data, the model undergoes rigorous testing, affirming its ability to provide accurate predictions and meaningful analyses. In this way, technical feasibility serves as a foundational pillar, ensuring that the proposed system is not only technically proficient but also capable of contributing valuable insights to the understanding and management of global health challenges.

3.3.3 Social Feasibility

Social feasibility, within the framework of developing a predictive risk model for forecasting future healthcare needs, involves a nuanced consideration of societal implications. Hospital and care home admissions, often unwelcome and costly events, share similarities, making them suitable for predictive modeling. The risk model's role in anticipating healthcare resource utilization is crucial for effective planning, allowing policymakers and healthcare providers to strategically allocate resources, thereby addressing societal needs and optimizing healthcare services.

The predictive model's application extends to the context of the coronavirus, enhancing social feasibility by forecasting infection rates. This proactive approach creates awareness, prompting individuals to adopt safety measures. By contributing to public health efforts, the model not only safeguards individuals but also supports broader societal well-being, preventing undue strain on healthcare systems. The integration of predictive modeling thus becomes a valuable tool in promoting collective resilience and community-wide efforts to combat the spread of infectious diseases.

3.4 System Specification

3.4.1 Hardware Specification

- Speed : 1.1GHz
- RAM: 32GB
- Storage : 300 GB
- Key Board : Standard Windows Keyboard
- Processor : Intel

3.4.2 Software Specification

- Operating System : Windows
- Technology : Machine Learning
- Tool: Google colab
- Version: 21H2

3.4.3 Standards and Policies

Google Colab

Google Colab, short for Colaboratory, is a cloud-based platform provided by Google that offers a similar environment to Jupyter Notebooks. While Anaconda provides an environment on your local machine, Google Colab allows users to run Jupyter notebooks directly in the cloud, comes with pre-installed libraries and supports popular machine learning frameworks like TensorFlow and PyTorch. Users can collaborate in real-time and leverage Google's powerful hardware, including GPUs, for accelerated computation. Google Colab also integrates seamlessly with Google Drive, making it convenient for data storage and sharing.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

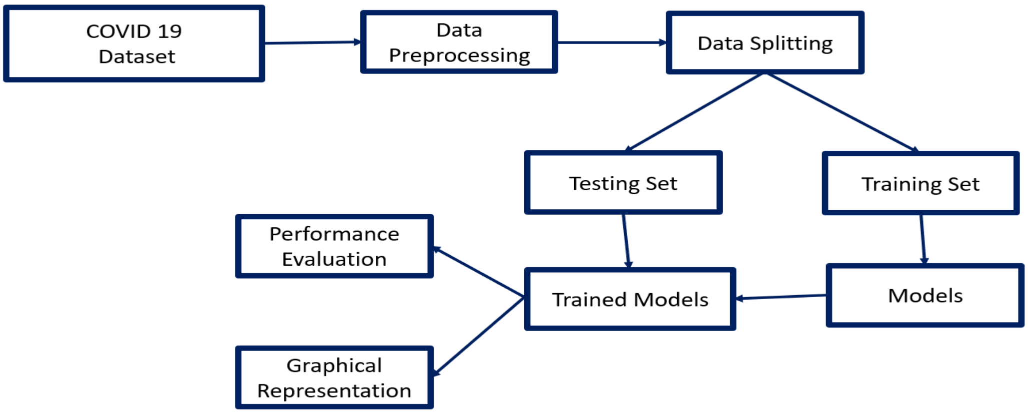

Figure 4.1: **Architecture Diagram**

Figure 4.1 illustrates the architectural diagram depicting the Time Series dataset sourced from John Hopkins University, USA, highlighting the initial step of partitioning the dataset into training and testing sets for effective machine learning model evaluation. LR, PR, and SVR are tested for their accuracy in predicting dataset trends. The standard machine learning process involves training models with historical data, testing on new data, and evaluating performance using a portion of available historical data, with accuracy as a key metric.The iterative approach ensures the robustness of our system, allowing it to effectively capture and forecast patterns within the Time Series dataset. The machine learning models are essential tools in discerning meaningful insights and trends from past data, contributing to accurate predictions and informed decision-making.

4.2 Design Phase

4.2.1 Data Flow Diagram

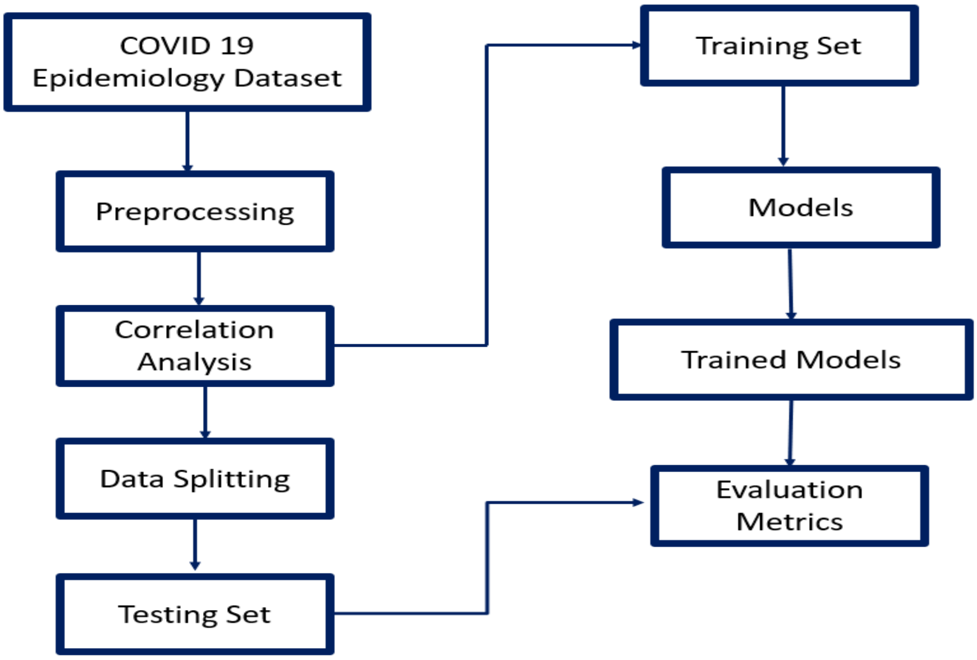

Figure 4.2: **Data Flow Diagram**

The Figure 4.2 describes a detailed flow diagram and guides the ML process for predicting COVID-19 infections. Flowchart starts with meticulous data collection, ensuring the acquisition of accurate and pertinent information. The collected data undergoes preprocessing to prepare for analysis. Subsequently, the dataset is split into training and test sets, facilitating effective model training and performance evaluation. The systematic approach ensures that the machine learning models are well-equipped to make accurate predictions tailored to the unique characteristics of the dataset.

4.2.2 Use Case Diagram

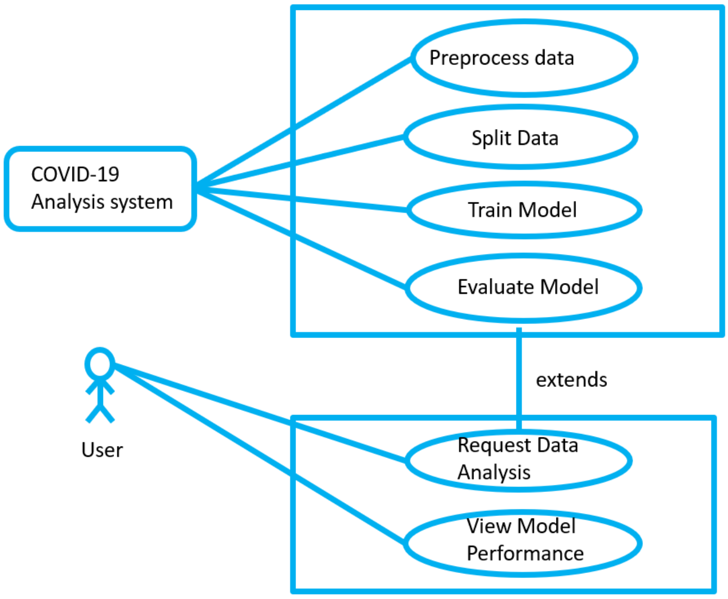

Figure 4.3: **Use Case Diagram**

The figure 4.3 illustrates the user interaction with the Covid-19 prediction system, where the SVM and polynomial feature models serve as its main components. The user-friendly interface allows individuals to input data and opt for a classification model build using the loaded dataset. This design ensures that users can seamlessly engage with the system, gaining insights into COVID-19 trends.At the system's essence is the capability to predict outcomes based on user inputs, employing advanced machine learning algorithms. The option to build a classification model enhances the system's versatility, adapting to varying datasets.

4.2.3 Sequence Diagram

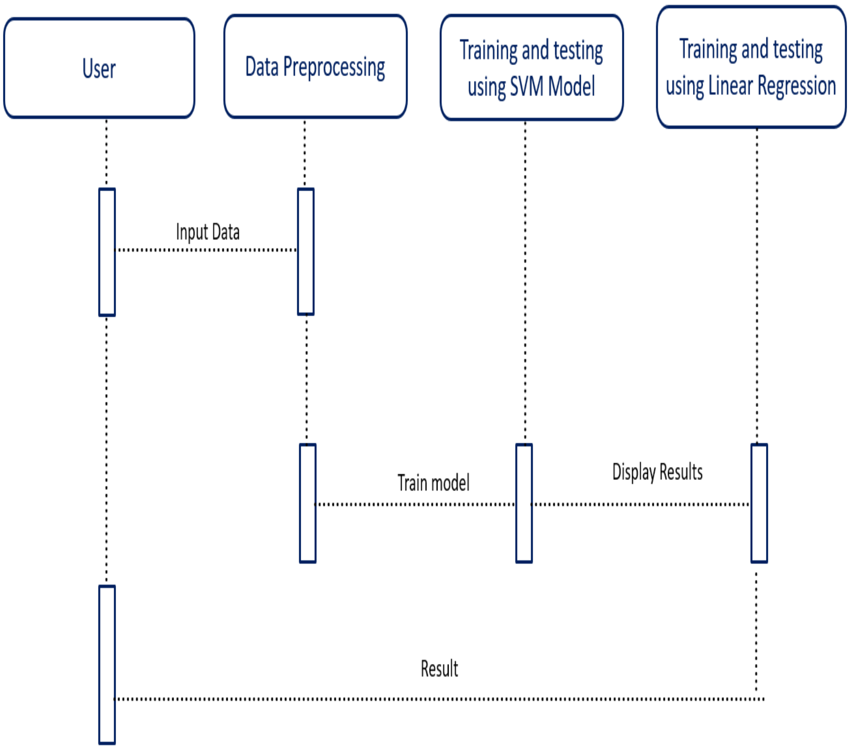

Figure 4.4: **Sequence Diagram**

In Figure 4.4, the sequence diagram commences with user initiation, triggering the system to gather COVID-19 testing data. This data is then subjected to a crucial preprocessing step, meticulously cleaning and transforming it for optimal analysis by machine learning models. The subsequent execution of tailored models, such as SVM or PR, unfolds, with these algorithms delving into the preprocessed data to discern patterns and relationships relevant to COVID-19 trends. The sequence reaches its conclusion by presenting the user with the analyzed results, offering valuable insights into the anticipated trajectory of the pandemic. Figure 4.5, through its systematic depiction, underscores the integrative and iterative nature of leveraging machine learning for in-depth COVID-19 analysis, encompassing data input, preprocessing, model execution, and result presentation.

4.2.4 Collaboration Diagram

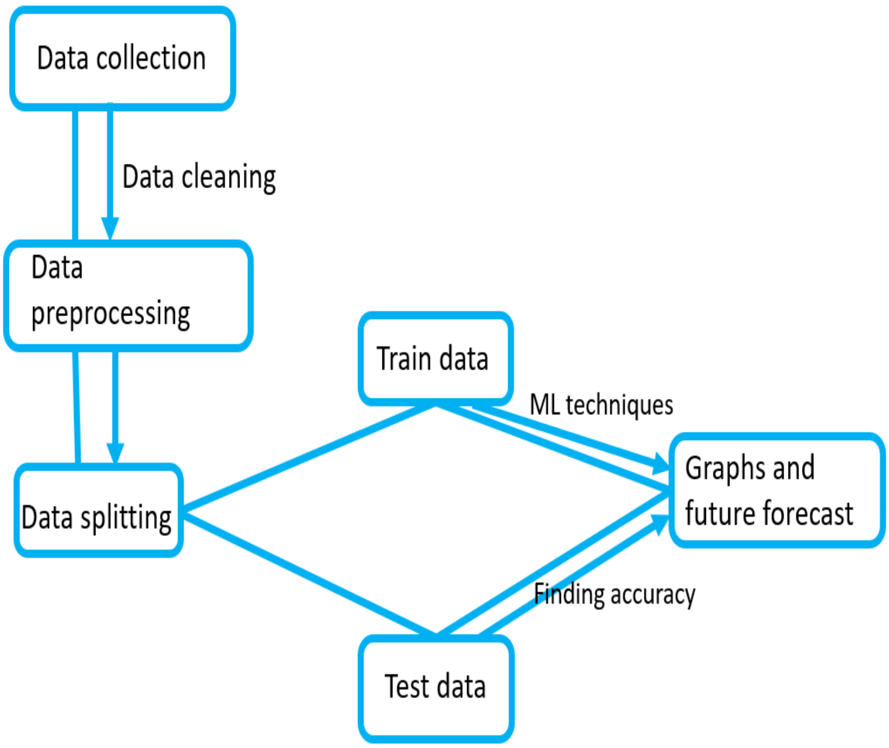

Figure 4.5: **Collabration Diagram**

In Figure 4.5, algorithms are intricately explained, detailing how simulation parameters are learned through supervised learning for predicting disease incidence. The diagram illustrates the dynamic process of adapting to evolving COVID-19 trends based on past data, showcasing the sophistication of our predictive models.The prediction level for COVID-19, employing time series forecasting models, introduces multiple objects.

4.2.5 Activity Diagram

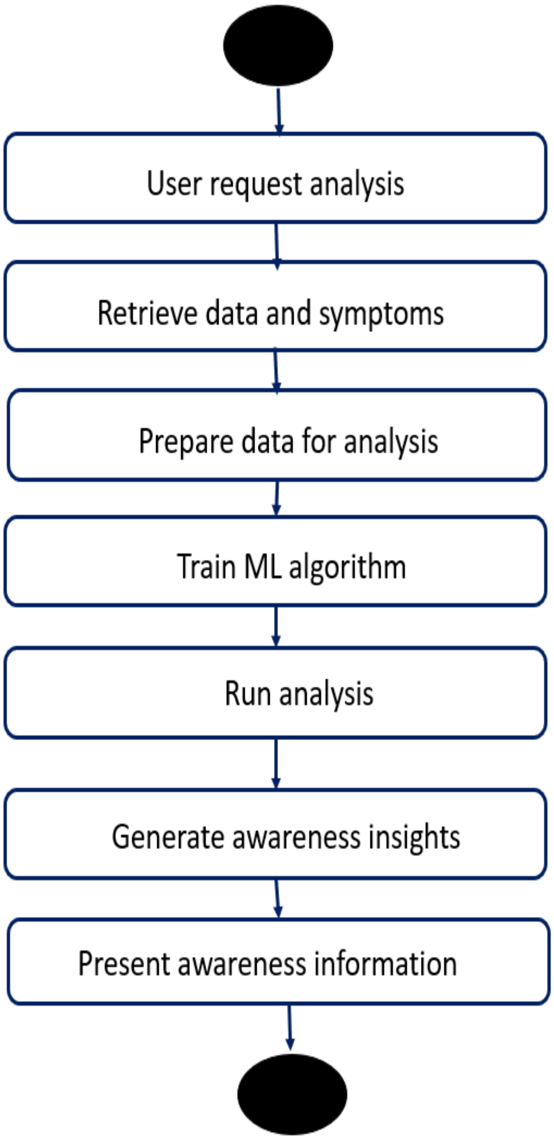

Figure 4.6: **Activity Diagram**

In Figure 4.6, the activity diagram for COVID-19 prediction outlines a series of essential steps from user request to generating awareness insights. Beginning with user request analysis, the process involves retrieving historical COVID-19 data and symptoms information, followed by data preparation through cleansing and preprocessing. Machine learning algorithms, such as Linear Regression or Support Vector Machines, are then trained on the prepared data. The analysis phase interprets the predictions, generating valuable awareness insights about the ongoing situation. Finally, the awareness information is presented to the user through comprehensible formats like charts or reports, providing actionable insights to increase understanding and vigilance regarding COVID-19 developments.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

1. Read COVID-19 dataset from the provided CSV file using ‘read csv‘.
2. Drop the unnecessary column ”SNo” and convert ”ObservationDate” to datetime.
3. Group data by ”ObservationDate,” summing up columns for ”Confirmed,” ”Recovered,” and ”Deaths.”
4. Print size/shape, check for null values, and display data types of each column.
5. Conduct various visualizations and analyses.
6. Calculate ”Days Since” the first observation and split data into training and validation sets.
7. Initialize Linear Regression model with normalization and SVM model with specified parameters.
8. Train both models using the training data.
9. Generate new dates for the next 17 days and predict COVID-19 cases for each day using both models.
10. Create a DataFrame with new dates and predictions, then display the first 5 rows.

4.3.2 Pseudo Code

```
# Read COVID-19 dataset
covid_data = read_csv("covid_19_data.csv")

# Preprocessing and exploration
covid_data.drop_column("SNo")
covid_data["ObservationDate"] = to_datetime(covid_data["ObservationDate"])
grouped_data = group_by(covid_data, ["ObservationDate"], sum_columns=["Confirmed", "Recovered", "
    Deaths"])

# Basic information and statistics
print("Size/Shape of the dataset", get_shape(covid_data))
print("Checking for null values", check_null_values(covid_data))
print("Checking Data-type", get_data_types(covid_data))

# Visualizations and analysis
# ... (bar plots, weekly progress, average daily increase, country-wise analysis, etc.)

# Machine learning model training and prediction
grouped_data["Days Since"] = calculate_days_since(grouped_data.index)
train_data = get_training_data(grouped_data)
validation_data = get_validation_data(grouped_data)

# Model initialization and training
```

```
linear_reg = initialize_linear_regression(normalize=True)
svm_model = initialize_svm(C=1, degree=5, kernel='poly', epsilon=0.001)
train_model(linear_reg, get_column_values(train_data, "Days Since"), get_column_values(train_data, "
    Confirmed"))
train_model(svm_model, get_column_values(train_data, "Days Since"), get_column_values(train_data, "
    Confirmed"))

# Model predictions and create a DataFrame
new_dates = []
predictions_lr = []
predictions_svm = []

for i in range(1, 18):
    new_dates.append(last_date(grouped_data) + timedelta(days=i))
    predictions_lr.append(predict(linear_reg, last_date(grouped_data) + i))
    predictions_svm.append(predict(svm_model, last_date(grouped_data) + i))

model_predictions = create_dataframe(new_dates, predictions_lr, predictions_svm, ["Dates", "LR", "
    SVR"])

# Display first 5 rows of the model predictions DataFrame
display(first_rows(model_predictions, 5))
```

4.4 Module Description

The goal is to predict virus trends over the next ten days by leveraging clinical data. Selecting an optimal ML methodology is crucial for precise predictions spanning various countries and regions. Within the scope of our COVID-19 prediction project, modules act as discrete units of functionality, systematically dividing the project into manageable components. Each module, such as "Dataset Collection," "Model Training," "System Integration," "Data Management and Update," and "Reporting and Visualization," focuses on specific functionalities, contributing to the overall predictive system. It is common for one module to depend on another, creating a structured and interdependent architecture. This modular organization streamlines code readability, facilitates maintenance, and supports scalability. By encapsulating related functionalities, modules enable efficient collaboration among developers and the reuse of code components, essential for managing the complexity inherent in our COVID-19 prediction project.

4.4.1 Dataset Collection

The dataset serves as the backbone of our COVID-19 prediction model, providing a rich and multidimensional set of information for training purposes. The ”Date” attribute allows the model to understand the temporal dynamics of the pandemic, crucial for capturing trends and patterns over time. Geographical details such as ”State” and ”District” enable the model to account for regional variations in the spread and impact of the virus. Metrics like ”Recovered,” ”Deceased,” and ”Tested” offer insights into the health outcomes and testing efforts, contributing to a holistic understanding of the pandemic's progression. The inclusion of ”Other” variables, if specified, adds flexibility to accommodate additional relevant information.

The dataset's availability on GitHub ensures transparency and facilitates collaboration within the global research community, aligning with the principles of open science. Its frequent updates allow the model to adapt to the dynamic nature of the pandemic, capturing the latest trends and variations. In essence, this dataset not only forms the foundational building blocks for our predictive model but also embodies a collaborative and adaptive approach to understanding and forecasting the complexities of the COVID-19 landscape.

4.4.2 Data preprocessing

In the realm of machine learning applications, preprocessing stands as a foundational and crucial step. The datasets encountered in machine learning endeavors are often unstructured and laden with noise. To pave the way for effective data analysis, it becomes imperative to transform the raw data into a structured and refined format. This is where data cleaning assumes a pivotal role, acting as the precursor to model building. The preprocessing phase involves a series of techniques applied to the dataset, preparing it for subsequent ML techniques.

The selection and preparation of the appropriate dataset are paramount in enhancing the efficacy of machine learning models. Often, dataset information is collected from diverse and disparate sources, necessitating a harmonization process to amalgamate them into a cohesive and standardized dataset format. Once this dataset preparation is completed, it is typically saved in the widely used .CSV format, ensuring ease of use in various ML environments.

The primary goal of preprocessing is twofold: to eliminate noise and extraneous data and to organize the dataset in a manner conducive to effective analysis. By cleansing the dataset of irrelevant information and inconsistencies, preprocessing enhances the quality and reliability of the data, setting the stage for robust machine learning model calculations. Subsequent application of machine learning algorithms on this refined dataset allows for meaningful and accurate predictions, classifications, or other analytical outcomes. In essence, preprocessing acts as a data refinement process, optimizing datasets for the optimal performance of machine learning models.

4.4.3 Training and testing data

The proposed model predicts the progression of the next ten days for a specific COVID-19 case type by extracting data from the preprocessed list within defined date intervals. Two distinct lists are then created, one for dates and another for the specified district name. It is employed to partition the dataset into training and testing sets, allowing for an effective evaluation of the machine learning model's performance on unseen data.

Following the data division, the model undergoes training using the training set. During this phase, the model learns from historical patterns and relationships within the data, adapting its internal parameters. Subsequently, the model's predictive capabilities are assessed using the testing set, representing data it has not encountered during the training process. This evaluation phase ensures that the model can generalize effectively to new instances, providing insights into its accuracy and reliability in forecasting COVID-19 trends for the specified case type.

4.5 Steps to execute/run/implement the project

4.5.1 step 1-Setup Environment

- Ensure Python is installed.
- Install required packages: matplotlib,numpy,scikit-learn,seaborn,statsmodels.
- Set up a working directory for the project.

4.5.2 Step 2 - Download Dataset

- Obtain the covid-19 dataset.
- Place the dataset in a folder accessible to the project

4.5.3 Step 3 - Project Structure

- Create a structured project folder.
- Include the script and a subfolder for the dataset.

4.5.4 Step 4 - Code Script

- Write the Python script with the required code.

4.5.5 Step 5 - Execute Code

- Run the script in a Python environment.
- This loads, preprocesses, and splits the dataset.

4.5.6 Step 6 - Training

- Train the model.
- specify number of Confirmed cases and number of Death cases.

4.5.7 Step 7 - Results and Analysis

- Display or log training and testing results, including accuracy.
- Analyze the model's performance and identify potential areas for improvement.

4.5.8 Step 8 - Save Model

- Save the trained model to a file.
- This enables reuse and deployment without retraining.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
0	1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1	0	0
1	2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14	0	0
2	3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6	0	0
3	4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1	0	0
4	5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0	0	0
5	6	01/22/2020	Guangdong	Mainland China	1/22/2020 17:00	26	0	0
6	7	01/22/2020	Guangxi	Mainland China	1/22/2020 17:00	2	0	0
7	8	01/22/2020	Guizhou	Mainland China	1/22/2020 17:00	1	0	0
8	9	01/22/2020	Hainan	Mainland China	1/22/2020 17:00	4	0	0
9	10	01/22/2020	Hebei	Mainland China	1/22/2020 17:00	1	0	0

Figure 5.1: **Covid-19 Metrics**

Figure 5.1 represents dataset of the early stages of the COVID-19 outbreak and provides a granular view of the virus's impact on different provinces or states within Mainland China. Analyzing the data over time or combining it with additional datasets could offer insights into the progression of the virus, regional variations, and the effectiveness of public health measures implemented during the early stages of the pandemic.

5.1.2 Output Design

	Dates	LR	SVR
0	2020-04-25	1560529	3322586
1	2020-04-26	1582219	3500761
2	2020-04-27	1603909	3686599
3	2020-04-28	1625599	3880344
4	2020-04-29	1647289	4082245

Figure 5.2: **Future Predictions using LR,SVR**

Figure 5.2 represents the output design describes the prediction for future dates using LR and SVR models. For each date in the next 17 days, the code appends the date to a list and predicts the number of cases using the LR and SVR models.The resulting predictions, along with their corresponding dates, are organized into a Pandas DataFrame named model predictions. This output serves as a concise representation of the forecasted values from both LR and SVR models, allowing for a comparative analysis of their predictions for the specified future dates.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Unit testing is a software testing technique where individual components or functions of a program are tested in isolation to ensure they behave as intended. It involves examining each unit's functionality independently of the rest of the system, helping identify and fix bugs early in the development process. Unit tests are typically automated and focus on specific aspects of code functionality.

Input

```
covid = pd.read_csv("covid_19_data.csv")
covid.head(10)
```

Test result

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
0	1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1	0	0
1	2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14	0	0
2	3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6	0	0
3	4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1	0	0
4	5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0	0	0
5	6	01/22/2020	Guangdong	Mainland China	1/22/2020 17:00	26	0	0
6	7	01/22/2020	Guangxi	Mainland China	1/22/2020 17:00	2	0	0
7	8	01/22/2020	Guizhou	Mainland China	1/22/2020 17:00	1	0	0
8	9	01/22/2020	Hainan	Mainland China	1/22/2020 17:00	4	0	0
9	10	01/22/2020	Hebei	Mainland China	1/22/2020 17:00	1	0	0

Figure 5.3: **Unit Testing**

The Figure 5.3 describes the correct loading of the dataset, handling of data types, and initial data exploration.Test cases could include checking if the ”ObservationDate” column is successfully converted to datetime format, confirming the absence of null values, and validating that the ”SNo” column is correctly dropped. Unit tests for these aspects would help ensure the integrity of the data processing steps within the code.

5.3.2 Integration testing

Integration testing is a software testing phase where individual components or modules of a system are combined and tested as a group to ensure they work seamlessly together. The primary goal is to detect any interactions or issues between integrated components, validating the system's overall functionality, data flow, and communication between different parts. Integration testing helps identify and address issues that may arise when combining individual units into a larger, cohesive system.

Input

```
covid["ObservationDate"] = pd.to_datetime(covid["ObservationDate"])
datewise = covid.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum
    "})
```

```
print("Basic Information")
print("Total number of Confirmed cases around the world",datewise["Confirmed"].iloc[-1])
print("Total number of Recovered cases around the world",datewise["Recovered"].iloc[-1])
print("Total number of Death cases around the world",datewise["Deaths"].iloc[-1])
print("Total number of Active cases around the world",(datewise["Confirmed"].iloc[-1]-datewise["
    Recovered"].iloc[-1]-datewise["Deaths"].iloc[-1]))
print("Total number of Closed cases around the world",(datewise["Recovered"].iloc[-1]+datewise["
    Deaths"].iloc[-1]))
```

Test result

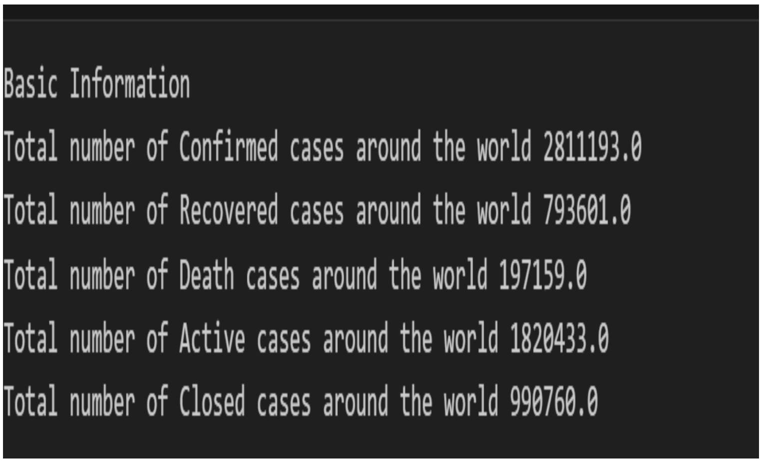

Figure 5.4: **Integration Testing**

The Figure 5.5 focus on verifying the seamless interaction between data preprocessing steps, ensuring accurate datetime conversion, and correct aggregation of COVID-19 cases.The test would validate the overall functionality of combining these operations to produce reliable global statistics. Integration testing aims to detect any inconsistencies or errors that may arise from the integration of these specific data processing components, guaranteeing the accuracy of the reported total confirmed, recovered, and death cases, as well as active and closed cases globally.

5.3.3 Functional testing

Functional testing is a software testing process that verifies individual functions and features of an application to ensure they meet specified requirements. It assesses the system's behavior, user interface, and data interactions, contributing to the overall reliability and quality of the software. This testing method aims to validate that the application functions correctly and aligns with its intended design and functionality.

Input

```
covid = pd.read_csv("covid_19_data.csv")
covid.drop(["SNo"],1,inplace=True)
covid.isnull().sum()
covid["ObservationDate"] = pd.to_datetime(covid["ObservationDate"])
```

5.3.4 Test Result

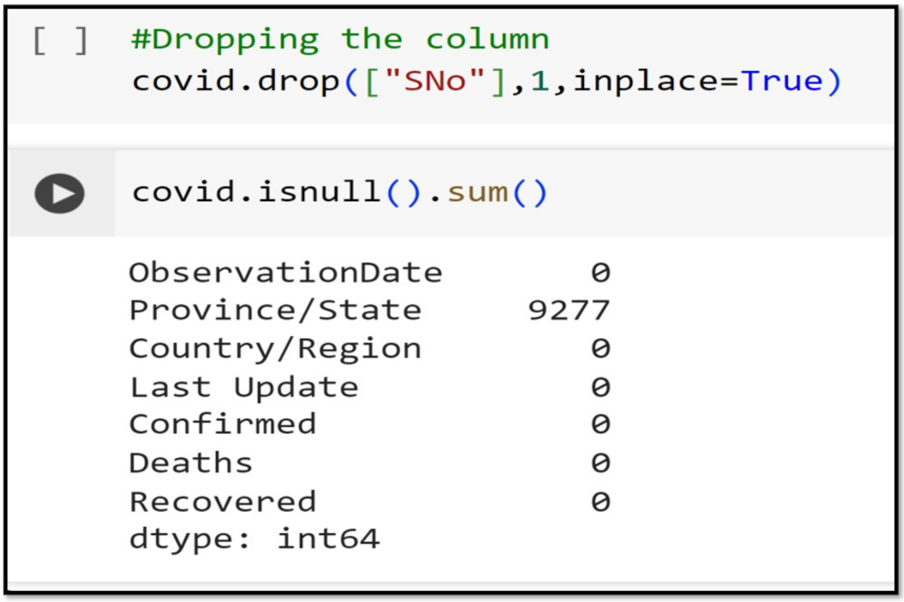

Figure 5.5: **Functional Testing**

The Figure 5.6, includes loading the dataset, managing null values, removing the "SNo" column, and converting "ObservationDate" to datetime format, are part of functional testing. Functional testing ensures that these data processing operations within the code function correctly and adhere to the expected behavior, contributing to the overall functionality and integrity of the software system.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed model for COVID-19 prediction and forecasting takes a comprehensive approach in response to the challenges posed by the pandemic. Initiated by the discourse on the efficacy of healthcare models, particularly patient-centered and hospital-centered approaches, the model seeks economically viable alternatives. In light of the escalating demand for healthcare resources, the discussion pivots towards exploring remotely managed healthcare delivery systems, aligning with the need for efficiency in resource allocation. The sets the stage for the integration of ML based approaches, where the model employs both SVM and Linear Regression algorithms.

SVM's role in the model is crucial, offering adaptability to high-dimensional and non-linear data patterns, excels in capturing complex relationships within COVID-19 datasets, identifying support vectors and optimizing hyperplanes. Simultaneously, Linear Regression contributes simplicity and interpretability, modeling linear relationships effectively. The model's efficiency emerges from the synergistic integration of these algorithms, aiming to provide accurate predictions and forecasts while offering insights into the multifaceted factors influencing the progression of the pandemic. The holistic strategy positions the proposed model as a valuable tool in navigating the complexities of COVID-19 and making informed decisions in the face of ongoing uncertainties.

6.2 Comparison of Existing and Proposed System

Existing system:

The existing healthcare models, primarily centered around patient and hospital-based approaches, have come under scrutiny due to their high costs. The global spread of COVID-19 has intensified the discourse on the need for more economically viable alternatives while maintaining healthcare service quality. The escalating demand for resources has prompted discussions within the medical community on achieving a balance without compromising patient care. In response to these challenges, there is a growing interest in exploring remotely managed healthcare delivery systems. While potentially applicable for patients with mild illnesses and no significant risk factors, the medical fraternity is actively evaluating the feasibility, benefits, and potential challenges of such systems to transition responsibly towards more sustainable and accessible healthcare practices.

Proposed system:

The proposed system presents a systematic approach to COVID-19 forecasting. It begins with model training using dedicated datasets, allowing the system to adapt to patterns in the data. The subsequent phases involve data manipulation, preprocessing, and feature extraction, ensuring accurate predictions. The system utilizes regression analysis models tailored for exposed Coronavirus cases to predict the disease spread rate for the next 7 days, driven by significant categorical variables. The iterative nature of the process ensures the model's adaptability to diverse datasets, providing valuable insights for proactive decision-making. The proposed system addresses the limitations of the existing model by incorporating systematic training, preprocessing steps, and enhanced predictive capabilities, thus offering a more robust and adaptable solution for managing the complexities of the COVID-19 pandemic.

Machine Learing Model	Training Dataset	Testing Dataset	F1-Score	Accuracy
SVM	14997	1703	0.97	0.97
KNN	4686	1563	0.87	0.95
DT Classifier	2028	518	0.90	0.94
Logistic Regression	354	99	0.84	0.88
Random Forest Regressor	240	60	0.87	0.87

Table 6.1: Comparision Table

The table 6.1 describes a comparative analysis of various machine learning models SVM, KNN, Decision Tree (DT) Classifier, Logistic Regression, and Random Forest Regressor evaluated on different-sized training and testing datasets. The F1-Score, representing the harmonic mean of precision and recall, provides insights into model performance, with SVM exhibiting the highest score at 0.97. Additionally, accuracy metrics are reported, with SVM achieving a remarkable 0.97 accuracy, indicating the overall effectiveness of the model on the testing dataset. The table underscores the importance of selecting appropriate models and dataset sizes for optimal classification performance.

6.3 Sample Code

```
covid = pd.read_csv("covid_19_data.csv")
covid.head(10)
print("Size/Shape of the dataset",covid.shape)
```

```
print("Checking for null values",covid.isnull().sum())
print("Checking Data-type",covid.dtypes)
covid.drop(["SNo"],1,inplace=True)
covid.isnull().sum()
covid["ObservationDate"] = pd.to_datetime(covid["ObservationDate"])
datewise = covid.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum
    "})
print("Basic Information")
print("Total number of Confirmed cases around the world",datewise["Confirmed"].iloc[-1])
print("Total number of Recovered cases around the world",datewise["Recovered"].iloc[-1])
print("Total number of Death cases around the world",datewise["Deaths"].iloc[-1])
print("Total number of Active cases around the world",(datewise["Confirmed"].iloc[-1]-datewise["
    Recovered"].iloc[-1]-datewise["Deaths"].iloc[-1]))
print("Total number of Closed cases around the world",(datewise["Recovered"].iloc[-1]+datewise["
    Deaths"].iloc[-1]))
plt.figure(figsize=(15,5))
sns.barplot(x=datewise.index.date,y=datewise["Confirmed"]-datewise["Recovered"]-datewise["Deaths"])
plt.title("Distributions plot for Active Cases")
plt.xticks(rotation=90)
plt.figure(figsize=(15,5))
sns.barplot(x=datewise.index.date,y=datewise["Recovered"]+datewise["Deaths"])
plt.title("Distribution plot for Closed Cases")
plt.xticks(rotation=90)
datewise["WeekofYear"] = datewise.index.weekofyear
week_num = []
weekwise_confirmed = []
weekwise_recovered = []
weekwise_deaths = []
w = 1
for i in list(datewise["WeekofYear"].unique()):
    weekwise_confirmed.append(datewise[datewise["WeekofYear"]==i]["Confirmed"].iloc[-1])
    weekwise_recovered.append(datewise[datewise["WeekofYear"]==i]["Recovered"].iloc[-1])
    weekwise_deaths.append(datewise[datewise["WeekofYear"]==i]["Deaths"].iloc[-1])
    week_num.append(w)
    w=w+1
plt.figure(figsize=(8,5))
plt.plot(week_num,weekwise_confirmed,linewidth=3)
plt.plot(week_num,weekwise_recovered,linewidth =3)
plt.plot(week_num,weekwise_deaths,linewidth = 3)
plt.xlabel("WeekNumber")
plt.ylabel("Number of cases")
plt.title("Weekly Progress of different types of cases")
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(12,4))
sns.barplot(x= week_num,y=pd.Series(weekwise_confirmed).diff().fillna(0),ax=ax1)
sns.barplot(x= week_num,y=pd.Series(weekwise_deaths).diff().fillna(0),ax=ax2)
ax1.set_xlabel("Week Number")
ax2.set_xlabel("Week Number")
ax1.set_ylabel("Numberof Confirmed cases")
ax2.set_ylabel("Numberof Death cases")
ax1.set_title("Weekly increase in number of Confirmed cases")
```

```
ax2.set_title("Weekly increase in number of Death Cases")
plt.show()
new_date = []
new_prediction_lr=[]
new_prediction_svm=[]
for i in range(1,18):
  new_date.append(datewise.index[-1]+timedelta(days=i))
  new_prediction_lr.append(lin_reg.predict(np.array(datewise["Days Since"].max()+i).reshape(-1,1))
      [0][0])
  new_prediction_svm.append(svm.predict(np.array(datewise["Days Since"].max()+i).reshape(-1,1))[0])
pd.set_option("display.float_format",lambda x: '%.f' % x)
model_predictions=pd.DataFrame(zip(new_date,new_prediction_lr,new_prediction_svm),columns = ["Dates"
    ,"LR","SVR"])
model_predictions.head(5)
```

Output

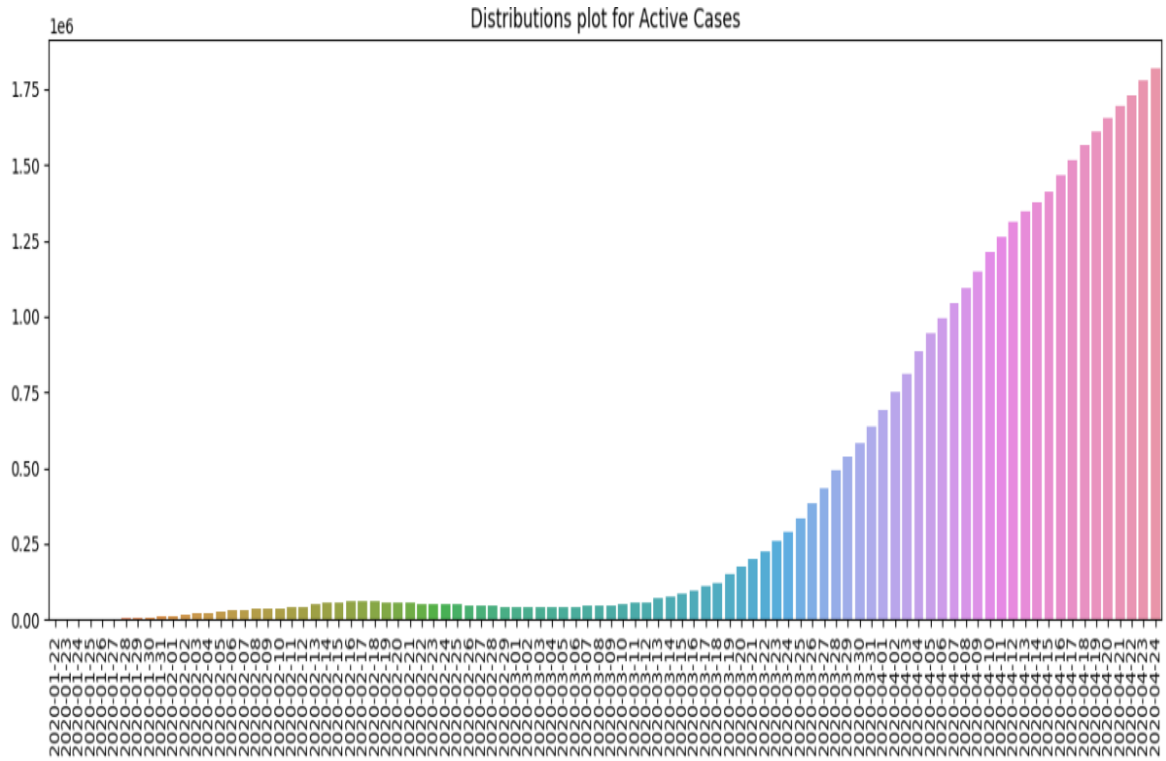

Figure 6.1: **Distribution Plot for Active Cases**

The figure 6.1 describes about the "Distribution plot for Active Cases," indicates its focus on showcasing the dynamics of active infections. The use of a bar plot allows for a clear visualization of how the number of active cases fluctuates over time.This visualization is valuable for observing trends and patterns in the progression of active COVID-19 cases, providing insights into the impact of interventions and public health measures over the specified timeframe.

	Dates	LR	SVR
0	2020-04-25	1560529	3322586
1	2020-04-26	1582219	3500761
2	2020-04-27	1603909	3686599
3	2020-04-28	1625599	3880344
4	2020-04-29	1647289	4082245

Figure 6.2: **Prediction of Covid-19 Cases**

The figure 6.2 describes the prediction for future dates using LR and SVR models. For each date in the next 17 days, the code appends the date to a list and predicts the number of cases using the LR and SVR models.The resulting predictions, along with their corresponding dates, are organized into a Pandas DataFrame named model predictions. This output serves as a concise representation of the forecasted values from both LR and SVR models, allowing for a comparative analysis of their predictions for the specified future dates.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

During the ongoing global pandemic, access to reliable data is crucial for disseminating tips and protective measures to safeguard individuals against the COVID-19 virus. To achieve a comprehensive analysis and summary of the COVID-19 dataset, data frames serve as a valuable tool. Leveraging ML techniques, particularly time series analysis, provides the capability to predict and forecast the trajectory of the pandemic in the coming ten days. The results obtained from the Pandemic Analyzer, built upon ML techniques, offer accurate insights into the virus's behavior. While forecasting may not achieve 100 percent accuracy, it serves as a valuable corrective measure, aiding in preparedness and response strategies.

In the pursuit of forecasting confirmed COVID-19 cases, a base model has been developed with a remarkable 93.74 percent accuracy in the prediction interval. This accuracy is achieved without any tweaking of seasonality-related parameters and additional regressors. The success of the base model underscores the potential of machine learning techniques in providing reliable and timely information for effective pandemic management.

7.2 Future Enhancements

In future endeavors, the Pandemic Analyzer will undergo upgrades through the incorporation of deep learning procedures, enhancing its capabilities to gather and process data from diverse datasets for the training of machine learning models. By utilizing advanced deep learning techniques, the analyzer aims to overcome limitations and refine its predictive accuracy. The focus will be on augmenting the system's efficiency and adaptability, ensuring it remains at the forefront of providing valuable insights during these challenging times.

Moreover, there is a recognition of the need for improvement in the current system. The models developed in this study, while successful in predicting confirmed cases, did not extend their capabilities to forecast the mortality rate. Predicting mortality rates is crucial for obtaining nuanced insights into the impact of the virus on a region. Future enhancements will address this gap, enabling the analyzer to deliver more comprehensive and accurate predictions, contributing to a more holistic understanding of the pandemic's impact on different regions.

Chapter 8

PLAGIARISM REPORT

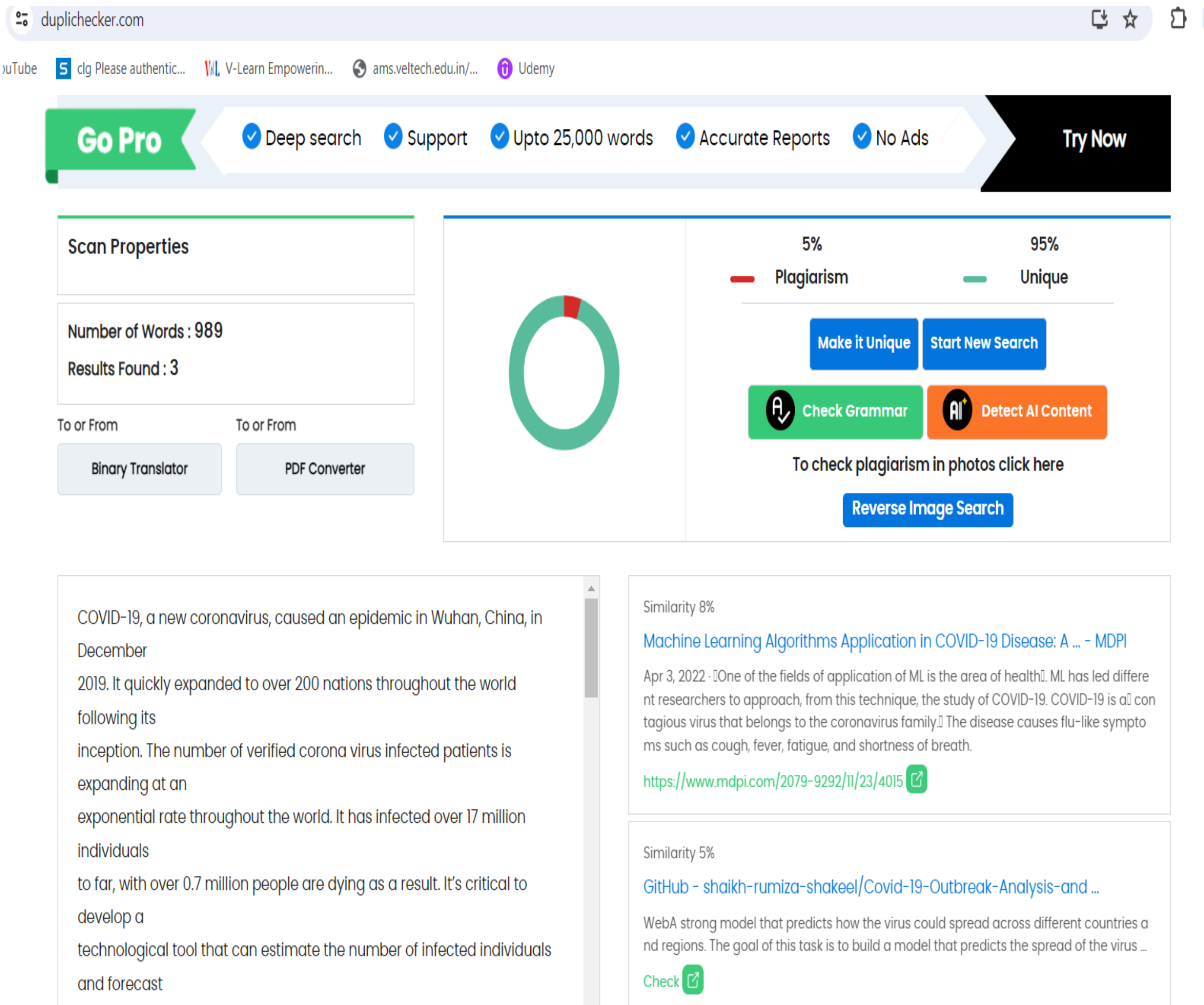

Figure 8.1: **Plagiarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import datetime as dt
from datetime import timedelta
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from statsmodels.tsa.api import Holt
covid = pd.read_csv("covid_19_data.csv")
covid.head(10)
print("Size/Shape of the dataset",covid.shape)
print("Checking for null values",covid.isnull().sum())
print("Checking Data-type",covid.dtypes)
covid.drop(["SNo"],1,inplace=True)
covid.isnull().sum()
covid["ObservationDate"] = pd.to_datetime(covid["ObservationDate"])
datewise = covid.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum
    "})
print("Basic Information")
print("Total number of Confirmed cases around the world",datewise["Confirmed"].iloc[-1])
print("Total number of Recovered cases around the world",datewise["Recovered"].iloc[-1])
print("Total number of Death cases around the world",datewise["Deaths"].iloc[-1])
print("Total number of Active cases around the world",(datewise["Confirmed"].iloc[-1]-datewise["
    Recovered"].iloc[-1]-datewise["Deaths"].iloc[-1]))
print("Total number of Closed cases around the world",(datewise["Recovered"].iloc[-1]+datewise["
    Deaths"].iloc[-1]))
plt.figure(figsize=(15,5))
sns.barplot(x=datewise.index.date,y=datewise["Confirmed"]-datewise["Recovered"]-datewise["Deaths"])
plt.title("Distributions plot for Active Cases")
plt.xticks(rotation=90)
plt.figure(figsize=(15,5))
sns.barplot(x=datewise.index.date,y=datewise["Recovered"]+datewise["Deaths"])
plt.title("Distribution plot for Closed Cases")
plt.xticks(rotation=90)
```

```
datewise["WeekofYear"] = datewise.index.weekofyear
week_num = []
weekwise_confirmed = []
weekwise_recovered = []
weekwise_deaths = []
w = 1
for i in list(datewise["WeekofYear"].unique()):
    weekwise_confirmed.append(datewise[datewise["WeekofYear"]==i]["Confirmed"].iloc[-1])
    weekwise_recovered.append(datewise[datewise["WeekofYear"]==i]["Recovered"].iloc[-1])
    weekwise_deaths.append(datewise[datewise["WeekofYear"]==i]["Deaths"].iloc[-1])
    week_num.append(w)
    w=w+1
plt.figure(figsize=(8,5))
plt.plot(week_num,weekwise_confirmed,linewidth=3)
plt.plot(week_num,weekwise_recovered,linewidth =3)
plt.plot(week_num,weekwise_deaths,linewidth = 3)
plt.xlabel("WeekNumber")
plt.ylabel("Number of cases")
plt.title("Weekly Progress of different types of cases")
fig,(ax1,ax2) = plt.subplots(1,2,figsize=(12,4))
sns.barplot(x= week_num,y=pd.Series(weekwise_confirmed).diff().fillna(0),ax=ax1)
sns.barplot(x= week_num,y=pd.Series(weekwise_deaths).diff().fillna(0),ax=ax2)
ax1.set_xlabel("Week Number")
ax2.set_xlabel("Week Number")
ax1.set_ylabel("Numberof Confirmed cases")
ax2.set_ylabel("Numberof Death cases")
ax1.set_title("Weekly increase in number of Confirmed cases")
ax2.set_title("Weekly increase in number of Death Cases")
plt.show()
print("Average increase in number of Confirmed cases everyday:",np.round(datewise["Confirmed"].diff
    ().fillna(0).mean()))
print("Average increase in number of Recovered cases everyday:",np.round(datewise["Recovered"].diff
    ().fillna(0).mean()))
print("Average increase in number of Death cases everyday:",np.round(datewise["Deaths"].diff().
    fillna(0).mean()))

plt.figure(figsize=(15,6))
plt.plot(datewise["Confirmed"].diff().fillna(0),label="Daily increase in confirmed cases",linewidth
    =3)
plt.plot(datewise["Recovered"].diff().fillna(0),label="Daily increase in recovered cases",linewidth
    =3)
plt.plot(datewise["Deaths"].diff().fillna(0),label="Daily increase in death cases",linewidth=3)
plt.xlabel("Timestamp")
plt.ylabel("Daily increase")
plt.title("Daily increase")
plt.legend()
plt.xticks(rotation=90)
plt.show()
countrywise= covid[covid["ObservationDate"]==covid["ObservationDate"].max()].groupby(["Country/
    Region"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum"}).sort_values(["Confirmed"],
```

```
    ascending=False)
countrywise["Mortality"]=(countrywise["Deaths"]/countrywise["Recovered"])*100
countrywise["Recovered"]=(countrywise["Recovered"]/countrywise["Confirmed"])*100
fig,(ax1,ax2)=plt.subplots(1,2,figsize=(25,10))
top_15confirmed = countrywise.sort_values(["Confirmed"],ascending=False).head(15)
top_15deaths = countrywise.sort_values(["Deaths"],ascending=False).head(15)
sns.barplot(x=top_15confirmed["Confirmed"],y=top_15confirmed.index,ax=ax1)
ax1.set_title("Top 15 countries as per number of confirmed cases")
sns.barplot(x=top_15deaths["Deaths"],y=top_15deaths.index,ax=ax2)
ax1.set_title("Top 15 countries as per number of death cases")
india_data = covid[covid["Country/Region"]=="India"]
datewise_india = india_data.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","
    Deaths":"sum"})
print(datewise_india.iloc[-1])
print("Total Active Cases",datewise_india["Confirmed"].iloc[-1]-datewise_india["Recovered"].iloc
    [-1]-datewise_india["Deaths"].iloc[-1])
print("Total Closed Cases",datewise_india["Recovered"].iloc[-1]+datewise_india["Deaths"].iloc[-1])
us_data = covid[covid["Country/Region"]=="US"]
datewise_us = us_data.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths"
    :"sum"})
print(datewise_us.iloc[-1])
print("Total Active Cases",datewise_us["Confirmed"].iloc[-1]-datewise_us["Recovered"].iloc[-1]-
    datewise_us["Deaths"].iloc[-1])
print("Total Closed Cases",datewise_us["Recovered"].iloc[-1]+datewise_us["Deaths"].iloc[-1])
datewise_india["WeekofYear"] = datewise_india.index.weekofyear
week_num_india = []
india_weekwise_confirmed = []
india_weekwise_recovered = []
india_weekwise_deaths = []
w = 1
for i in list(datewise_india["WeekofYear"].unique()):
    india_weekwise_confirmed.append(datewise_india[datewise_india["WeekofYear"]==i]["Confirmed"].
        iloc[-1])
    india_weekwise_recovered.append(datewise_india[datewise_india["WeekofYear"]==i]["Recovered"].
        iloc[-1])
    india_weekwise_deaths.append(datewise_india[datewise_india["WeekofYear"]==i]["Deaths"].iloc[-1])
    week_num_india.append(w)
    w=w+1
plt.figure(figsize=(8,5))
plt.plot(week_num_india,india_weekwise_confirmed,linewidth=3)
plt.plot(week_num_india,india_weekwise_recovered,linewidth =3)
plt.plot(week_num_india,india_weekwise_deaths,linewidth = 3)
plt.xlabel("WeekNumber")
plt.ylabel("Number of cases")
plt.title("Weekly Progress of different types of cases")
max_ind = datewise_india["Confirmed"].max()
china_data = covid[covid["Country/Region"]=="Mainland China"]
Italy_data = covid[covid["Country/Region"]=="Italy"]
US_data = covid[covid["Country/Region"]=="US"]
spain_data = covid[covid["Country/Region"]=="Spain"]
```

```
datewise_china = china_data.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum"})
datewise_Italy = Italy_data.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum"})
datewise_US=US_data.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum"})
datewise_Spain=spain_data.groupby(["ObservationDate"]).agg({"Confirmed":"sum","Recovered":"sum","Deaths":"sum"})
print("It took",datewise_india[datewise_india["Confirmed"]>0].shape[0],"days in India to reach",max_ind,"Confirmed Cases")
print("It took",datewise_Italy[(datewise_Italy["Confirmed"]>0)&(datewise_Italy["Confirmed"]<=max_ind)].shape[0],"days in Italy to reach number of Confirmed Cases")
print("It took",datewise_US[(datewise_US["Confirmed"]>0)&(datewise_US["Confirmed"]<=max_ind)].shape[0],"days in US to reach number of Confirmed Cases")
print("It took",datewise_Spain[(datewise_Spain["Confirmed"]>0)&(datewise_Spain["Confirmed"]<=max_ind)].shape[0],"days in Spain to reach number of Confirmed Cases")
print("It took",datewise_china[(datewise_china["Confirmed"]>0)&(datewise_china["Confirmed"]<=max_ind)].shape[0],"days in China to reach number of Confirmed Cases")
datewise["Days Since"]=datewise.index-datewise.index[0]
datewise["Days Since"] = datewise["Days Since"].dt.days
train_ml = datewise.iloc[:int(datewise.shape[0]*0.95)]
valid_ml = datewise.iloc[:int(datewise.shape[0]*0.95):]
model_scores=[]
lin_reg = LinearRegression(normalize=True)
svm = SVR(C=1,degree=5,kernel='poly',epsilon=0.001)
lin_reg.fit(np.array(train_ml["Days Since"]).reshape(-1,1),np.array(train_ml["Confirmed"]).reshape(-1,1))
svm.fit(np.array(train_ml["Days Since"]).reshape(-1,1),np.array(train_ml["Confirmed"]).reshape(-1,1))
prediction_valid_lin_reg = lin_reg.predict(np.array(valid_ml["Days Since"]).reshape(-1,1))
prediction_valid_svm = svm.predict(np.array(valid_ml["Days Since"]).reshape(-1,1))
new_date = []
new_prediction_lr=[]
new_prediction_svm=[]
for i in range(1,18):
    new_date.append(datewise.index[-1]+timedelta(days=i))
    new_prediction_lr.append(lin_reg.predict(np.array(datewise["Days Since"].max()+i).reshape(-1,1))[0][0])
    new_prediction_svm.append(svm.predict(np.array(datewise["Days Since"].max()+i).reshape(-1,1))[0])
pd.set_option("display.float_format",lambda x: '%.f' % x)
model_predictions=pd.DataFrame(zip(new_date,new_prediction_lr,new_prediction_svm),columns = ["Dates","LR","SVR"])
model_predictions.head(5)
```

9.2 Poster Presentation

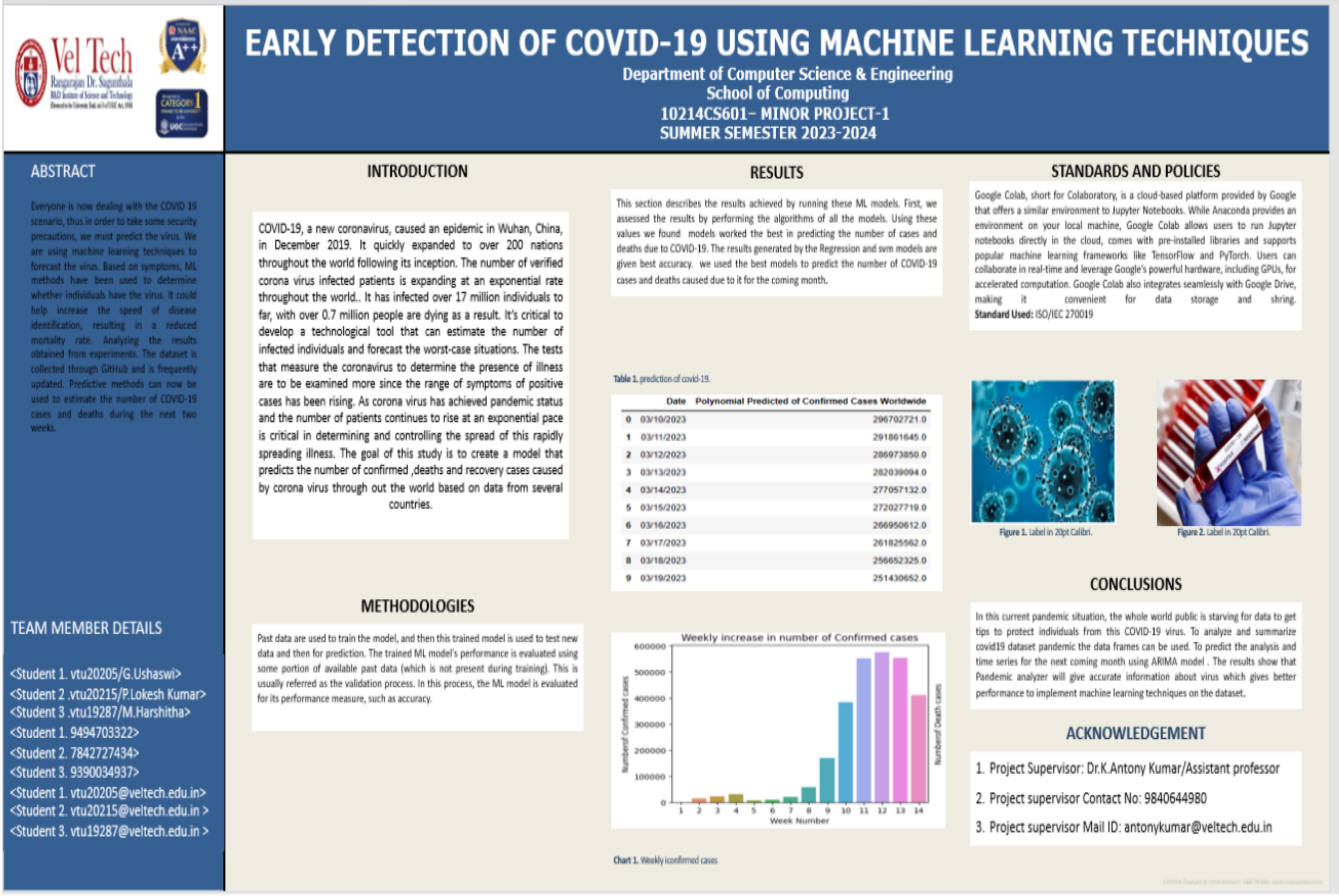

Figure 9.1: **Poster**

References

[1] Alazab, M., Awajan, A., Mesleh, A., Abraham, A., Jatana, V. and Alhyari, S., 2020. COVID-19 prediction and detection using deep learning. International Journal of Computer Information Systems and Industrial Management Applications, 12(June), pp.168-181.

[2] Arora, P., Kumar, H. and Panigrahi, B.K., 2020. Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India. Chaos, solitons fractals, 139, p.110017.

[3] Parbat, D. and Chakraborty, M., 2020. A python based support vector regression model for prediction of COVID19 cases in India. Chaos, Solitons Fractals, 138, p.109942.

[4] Rath, S., Tripathy, A. and Tripathy, A.R., 2020. Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model. Diabetes metabolic syndrome: clinical research reviews, 14(5), pp.1467-1474.

[5] Singh, V., Poonia, R.C., Kumar, S., Dass, P., Agarwal, P., Bhatnagar, V. and Raja, L., 2020. Prediction of COVID-19 corona virus pandemic based on time series data using Support Vector Machine. Journal of Discrete Mathematical Sciences and Cryptography, 23(8), pp.1583-1597.

[6] Ahmad, F., Almuayqil, S.N., Humayun, M., Naseem, S., Khan, W.A. and Junaid, K., 2021. Prediction of COVID-19 cases using machine learning for effective public health management. Comput. Mater. Contin, 66, pp.2265-2282.

[7] Guhathakurata, S., Kundu, S., Chakraborty, A. and Banerjee, J.S., 2021. A novel approach to predict COVID-19 using support vector machine. In Data Science for COVID-19 (pp. 351-364). Academic Press.

[8] Khodeir, M.M., Shabana, H.A., Alkhamiss, A.S., Rasheed, Z., Alsoghair, M., Alsagaby, S.A., Khan, M.I., Fernandez, N. and Al Abdulmonem, W., 2021. Early prediction keys for COVID-19 cases progression: A meta-analysis. Journal of infection and public health, 14(5), pp.561-569.

[9] Tiwari, S., Chanak, P. and Singh, S.K., 2022. A review of the machine learning algorithms for COVID-19 case analysis. IEEE Transactions on Artificial Intelligence.

[10] Gothai, E., Thamilselvan, R., Rajalaxmi, R.R., Sadana, R.M., Ragavi, A. and Sakthivel, R., 2023. Prediction of COVID-19 growth and trend using machine learning approach. Materials Today: Proceedings, 81, pp.597-601.