Research Lab

AI and Covid

Date of Submission: 13th January 2023

Participant name	E-mail Address	Matriculation Number	Course of study
Md Naiem Siddique	msiddique@uni-koblenz.de	219203013	M.Sc Web and Data Science
Martin Ebuka Okolie	mokolie@uni-koblenz.de	219203259	M.Sc Web and Data Science
Rahul Bhanushali	rbhanushali@uni-koblenz.de	221100681	M.Sc Web and Data Science
Pathey Atulkumar Pandya	patheypandya@uni-koblenz.de	219203197	M.Sc Web and Data Science
Boyon Dey Shipon	boyondey@uni-koblenz.de	219203319	M.Sc Web and Data Science
Rahul Chhabadiya	rchhabadiya@uni-koblenz.de	219203207	M.Sc Web and Data Science

Universität Koblenz
Supervisors: Prof. Dr. Maria Wimmer, Dr. Ulf Lotzmann
Koblenz

Table of figures

Abstract

Author: Md Naiem Siddique

The first human infection of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) was reported in December 2019 in Wuhan, China. Ever since it has infected over 630 million people and caused over 6.5 million deaths worldwide until early November 2022 (data source: WHO). Governments, civil protection departments, healthcare organizations, citizens and other concerned offices has been working together against this pandemic to contain the virus. Different preventive measures such as vaccination, lockdown, isolation, mask usage, social distancing, remote work etc. has shown positive outcomes over the spread and effects of the virus (Saha et al., 2022) Additionally, Inherent social, economic and geospatial aspects such as demographics, population density, income, gender, age, social structure etc. influences the reproduction of COVID infections (Sy et al., 2021) (Cao & Heydari, 2022) (Davies et al., 2020). The objective of our research is to better understand the situation based on covid related official report data and to support decision maker making effective decisions.

We aim to capture the periodic COVID-19 report by structuring and storing them in a persistent storage. Additionally, we intend to monitor the progression of the disease. By using machine learning algorithms, we will forecast the number of COVID cases of recent future. The primary subject of our research is limited to 36 German cities of state Rhineland-Platinate. An interactive visual representation of the collected data will assist the stakeholders to observe the recent and future trend of infection. User of our system will be able to answer set of questions to analyse the cause and effect of the COVID virus.

Introduction

Author: Martin Ebuka Okolie

Artificial intelligence (AI) has attracted a lot of attention in recent years, and the current COVID-19 epidemic has only served to emphasise how useful AI may be in resolving global issues. In this study, we investigate the effect of AI in combating the COVID-19 epidemic, especially in the German state of Rhineland-Palatinate.

Every sector of society has been significantly impacted by the COVID-19 epidemic, including the economy, healthcare, and daily life. AI has the potential to significantly contribute to reducing the pandemic's effects and enhancing the quality of life for residents of the Rhineland-Palatinate area.

This research project aims to assess the potential use of AI in addressing the COVID-19 pandemic in the Rhineland Palatinate region. By using previous COVID-19 data to make forecasts, this provides the region with information about how the risk of infection may be in the nearest future, hereby reducing the risk and aiding the healthcare sector, economic recovery, and daily life. We also hope to identify opportunities for AI to make a positive impact in the region.

Overall, this research project aims to provide a comprehensive analysis and attempt to forecast COVID-19 infection, hospitalization, and deceased rate in order to play a role in addressing the COVID-19 pandemic in the Rhineland Palatinate region and to offer recommendations for the effective implementation of government policies.

Project Management

Author: Md Naiem Siddique

Team Organization

- **Project Leaders**

Team Leaders	From	To
Boyon Dey Shipon	9th May 2022	30th May 2022
Rahul Bhanushali	31st May 2022	11th July 2022
Martin Ebuka Okolie	12th July 2022	26th July 2022
Pathey Atulkumar Pandya	27th July 2022	30th August 2022
Md Naiem Siddique	31st August 2022	18th September 2022
Rahul Chhabadiya	19th September 2022	28th October 2022
Martin Ebuka Okolie	29th October 2022	28th November 2022

Figure 1 Project Leaders

- **Reporting**

Project Managers
Prof. Dr. Maria A. Wimmer
Dr. Ulf Lotzmann

Project meetings

- **Formal/External Meetings**

#	Presentation topics	Feedbacks	Dates
0	• Initial kick off presentation • Team introduction • Primary project plan	• Study expert systems • Find relevant and reliable data sources. • Find more teammates	22/04/2022
1	• Final research questions • Data dimension and structure • Project plan • Team & project management	• Consider Rhineland-Pfalz as test region and use official dataset from lua.rpl.de. • Use “Open project” for collaboration and project manage activities. • Analyse Influencing factor of COVID infection.	9/05/2022
2	• Data profiling, parameter analysis • Data schema design • Database and visualization technology exploration.	• Define deliverables such as work packages, milestones in Open Project. • Create software module architecture. • Data visualization demonstration on Tableau.	31/05/2022
3	• Project plan • Project Management Dashboard • Database Schema • Data Visualization Demo with Tableau • Module Architecture	• Work package and time estimation • Create data archive in cloud. • Explore map-view in visualization. • Update and finalize database schema. • Collect historic data and mitigate inconsistencies.	12/07/ 2022
4	• Data pre-processing • Database and schema execution	• Prepare schema description document. • Set milestones. • Explore Gantt chart.	27/07/ 2022
5	• Historical data storing • Machine learning model exploration	• Create data pipeline. • Explore ARIMA model. • Fix geolocation data issue	31/08/2022
6	• Trend analysis • Visualization demonstration	• Implement ML Model • Use weekly data for trend analysis	19/09/2022
7	• Data Pipeline Automation • Logger • ML Model Implementation	• Implement Smoothing • Consider moving average	29/10/2022
8	• Final Presentation • Results and findings demonstration	• Define Future Scope • Incorporate additional factors	29/11/2022

- **Internal Meetings**

We have arranged weekly team meetings on every week on Monday. The meetings took around 30 minutes to 1 hour each. We followed the scrum ritual to organize our meetings. Each of the team members had to discuss mainly three topics in general: 1. What we have done last week 2. What are the tasks for this week and 3. Is there any challenges or dependencies? Nevertheless, individual one to one or cross team meetings was out of the scope of this weekly meeting. Separate sessions involving the personnel has been arranged during our weekly scrum meeting.

Project plan

Work packages

WP Number	Work Package Name	Milestone Number
1	Project Management	2,9
2	Project Planning	1
3	Architecture Development	3,4
4	System Development	5,6,7
5	Integration and Testing	8

Milestones

Milestone Number	Milestone Description	Estimated Completion Date	Means of Verification
1	Project plan completed	30th August 2022	Project plan
2	Project requirements established	12th July 2022	Project plan
3	Data profiling completed	12th July 2022	Parameter analysis
4	Data collection successful	31st August 2022	MySQL database
5	Data collection pipeline automated	29th October 2022	MySQL database
6	Machine learning model executed	29th October 2022	Notebook
7	Dashboard created	19th September 2022	Dashboard
8	Functional and integration testing done	18th November 2022	GitHub
9	Final project handbook available	12th January 2023	Project Handbook

Literature Review

Author: Rahul Bhanushali

It is crucial to have a precise understanding of the spread of COVID-19 in order to effectively contain the virus and allocate resources appropriately. Artificial intelligence (AI) and machine learning (ML) have been widely applied in efforts to combat the COVID-19 pandemic. (Islam et al., 2021) There have been few comprehensive literature reviews conducted to summarize the current knowledge and identify future research directions in this area. Previous studies have focused on the use of data science in COVID-19 research. The authors of [explaining COVID-19] used AI and ML methods to predict and analyse the disease, including estimating the number of infected individuals, the rate of spread, and the effectiveness of model-based parameter estimation methods using county-level data on cases and deaths (Menda et al., 2021). This review aims to provide a methodology for fitting these models to available data.

The Susceptible-Infectious-Recovered (SIR) model, with various modifications, has been frequently used in the analysis of COVID-19. (Srivastava et al., 2020) This model is a standard model for understanding disease spread, in which individuals can be classified as susceptible (S), exposed (E), infected (I), recovered (R), or deceased (D). (Menda et al., 2021) The SEIRD model, which includes an "exposed" class, is a variant of the SIR model. Previous literature has used these models to analyse the spread of COVID-19 and inform public health responses. (Srivastava et al., 2020)

In the paper (Menda et al., 2021), the Covid-19 data from the United States of America was used to analyse the diverse effects in the US. On the other hand, the paper (Spannaus et al., 2022a) uses Covid-19 cases count data with respect to metropolitan US cities namely New York and Tennessee. For this research lab, we are focusing on Covid-19 data for the Rhineland Palatinate region of Germany consisting of the number of cases of infection, hospitalization, and deceased.

Authors of (Spannaus et al., 2022b) generated time-varying and time-constant reports to provide predictions of Covid cases in the future. Here, the window of one week was used to forecast the predictions for the future. It was found that both reports provided different results and accuracy. It is important to find the optimal window of the time period to generate forecasts in the future for maximum accuracy and prevent overfitting of predictive results.

Project Scope

Author: Md Naiem Siddique

1. Collection of COVID related historic data such as number of infection (new, total, and weekly), recovery, and hospitalization of Rhineland Palatinate and its districts. The official data sources will from be state statistical official (Landesuntersuchungsamt) of Rhineland Palatinate state in Germany.
2. Creation of a database and schema to store the historic data. A supporting software application interface has to be developed to periodically update the database with latest dataset.
3. Analysis of the dataset and investigating key factors (vaccination rate, demographics, age, health condition, social connectedness, interventions, occupation etc.) which might influence the rate of infection.
4. Development of Machine Learning algorithm to forecast the trend of infection for the upcoming weeks.

Research Questions

1. What is the trend of COVID-19 infection in different cities of Rhineland Palatinate state over a selected (past) time interval?
2. How might the infection trend look (forecast) in the following weeks?

Project Components

Author: Rahul Bhanushali

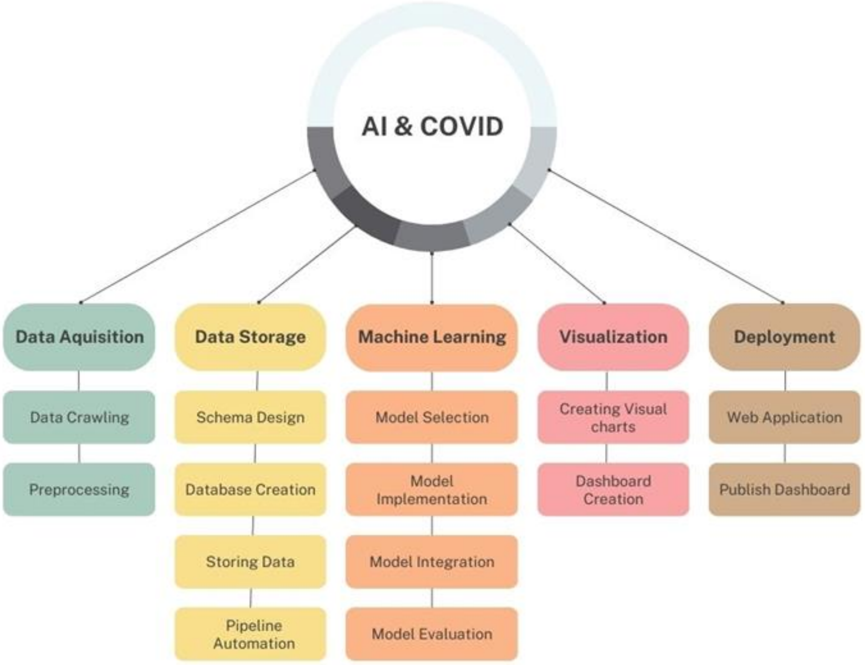


Figure 2 Project Components

Tools and Technology selection

Author: Martin Ebuka Okolie and Pathey Atulkumar Pandya

Python is a widely used programming language for data analysis and machine learning activities. It includes several tools and frameworks that make it simple to retrieve data from a variety of sources, create machine learning algorithms, and do statistical analysis. As a result, it's a useful tool for dealing with data and building machine-learning models.

MySQL is a database management system that is widely used for data storage. It enables the effective storing and retrieval of massive volumes of structured data and provides a variety of tools for data management and querying. As a result, it is a valuable tool for storing data utilized in data analysis and machine learning activities.

Streamlit and **Tableau** are both tools for data visualization and displaying findings. Tableau is a standalone software package intended exclusively for data visualization and dashboard development, whereas Streamlit is a Python library that facilitates the building of interactive web applications for data visualization. Both tools provide a variety of choices for visualizing data and showing findings in a clear and informative manner, which is critical for comprehending and sharing the outcomes of data analysis and machine learning activities.

Architecture Development

System Architecture

Author: Md Naiem Siddique

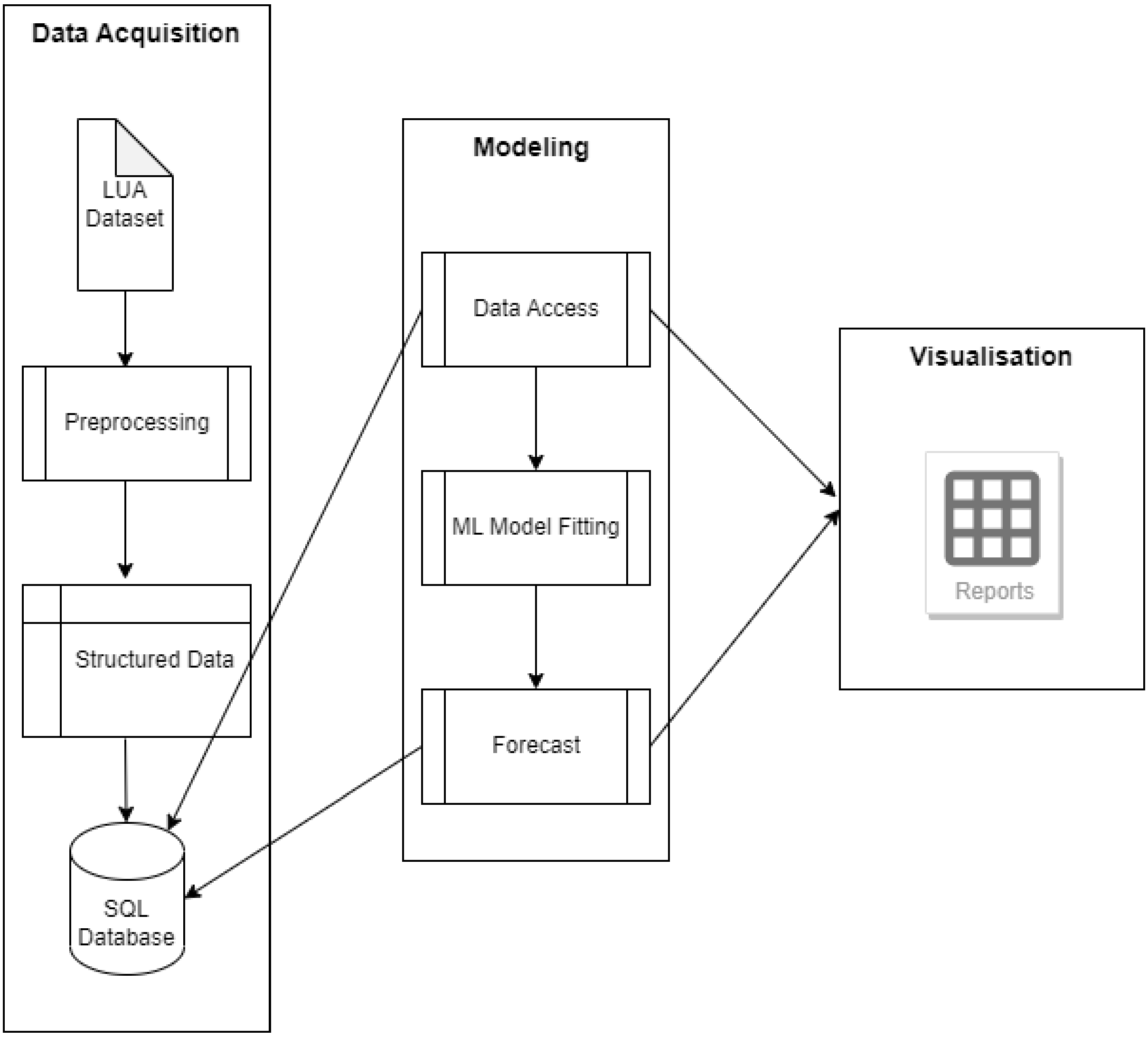


Figure 3 AI and COVID System Architecture

Data Architecture

Author: Rahul Bhanushali

Data Source selection

Data source selection is an important consideration in any machine learning project because the quality and relevance of the data can have a significant impact on the accuracy and effectiveness of the model. Some key factors to consider when selecting a data source for a machine learning project include:

1. **Relevance**: The data should be relevant to the problem or task that the model is being designed to solve.
2. **Quality**: The data should be accurate, consistent, and free from errors.
3. **Quantity**: There should be sufficient data to train and validate the model.
4. **Diversity**: The data should be diverse enough to accurately represent the range of possible inputs and outputs that the model will encounter in real-world use.
5. **Compatibility**: The data should be compatible with the tools and technologies being used to develop the model.

6. **Accessibility**: The data should be readily accessible and legally permitted for use in the project.

Considering the above factors, Covid-19 data is fetched from Homepage Landesuntersuchungsamt (rlp.de) for the region of Rhineland Palatinate, Germany as the data is timely updated and is legally accessible to use in the project. There are multiple files of data on the above-mentioned website. However, data is mainly acquired from two specific excel files on the website: (a) Daily Cases Report and (b) Weekly cases report.

1. Daily Cases Report: These report files are updated on daily basis and consist of fields such as city names within the Rhineland Palatinate region, number of cases for recovered, deceased, hospitalized, and new cases.
2. Weekly Cases Report: These report files are updated every Thursday on weekly basis. This report consists of the rate of hospitalization, deceased, and infection for a specific city within the Rhineland Palatinate region with the bifurcation of age groups. Age groups are divided into four categories: 0-11, 12-19, 20-59, and above 60.

Data Extraction using Web Crawler

Authors: Rahul Chhabadiya

Python is one of the best programming languages for web data crawling and analysis. With its vast array of libraries, it is an ideal language for extracting and processing data from the web.

Data crawling is the process of automatically collecting data from the web. Data crawlers can be used to locate and extract data from web pages and other sources to build a database of useful information. This can be useful for research, data analysis, and creating applications that require web data.

Python offers a wide variety of libraries for data crawling and data analysis. Some of the most popular libraries include Scrapy, BeautifulSoup, Requests, Pandas, NumPy and SciPy. These libraries provide powerful APIs for interacting with the web, making it easy to extract data, analyse and visualize.

Web Crawling

Historic data of covid-19 cases of Rhineland-Pfalz is crawled using python from this website. 'https://lua.rlp.de/de/unsere-themen/infektionsschutz/meldedaten-coronavirus/meldedaten-excel/'.

The process begins by making a request to the website and using the BeautifulSoup library to parse the response. It then creates a directory called 'coviddailydata' for daily data, ‘covidweeklydata’ for weekly data and saves the files from the website in those directories as the raw data without any manipulation. After that, it uses the pandas library to read the files in the directory and create a dataframe. Every time when the code run it check directories and existing files so that it does not get overwritten.

Data Pre-Processing & Storing

Author: Rahul Chhabadiya (90%) and Martin Ebuka Okolie (10%)

While loading the data in dataframe, data is cleaned and usable as per the decided data schema. The raw data contains the details which are not useful. The data frames are concatenated into one data frame and a few unwanted columns are dropped from the data frame and one column is added.

A database connection is established using the mysql.connector module and the data is inserted into the database using a for loop and the cursor object. The data is checked for duplication, and if no duplicate data is found, the data is inserted into the database. Finally, the connection to the database is closed.

This entire process is designed to crawl, clean, and store all the historic data as well as the upcoming data as per the data schema.

The dataset contains inconsistent. Initial data is consistent meaning the data is provided every day without omission. But from “30/04/2022” data is inconsistent. The data is not published on weekends and public holidays. To overcome that issue, in visualization, the data is used on weekly basis to overcome dramatic fluctuations. To overcome that issue in machine learning process, data smoothing process is much more realistic. The dataset contained missing values as shown in the figure below, but this is addressed in the Machine learning section of this paper under *Current Trend Analysis*.

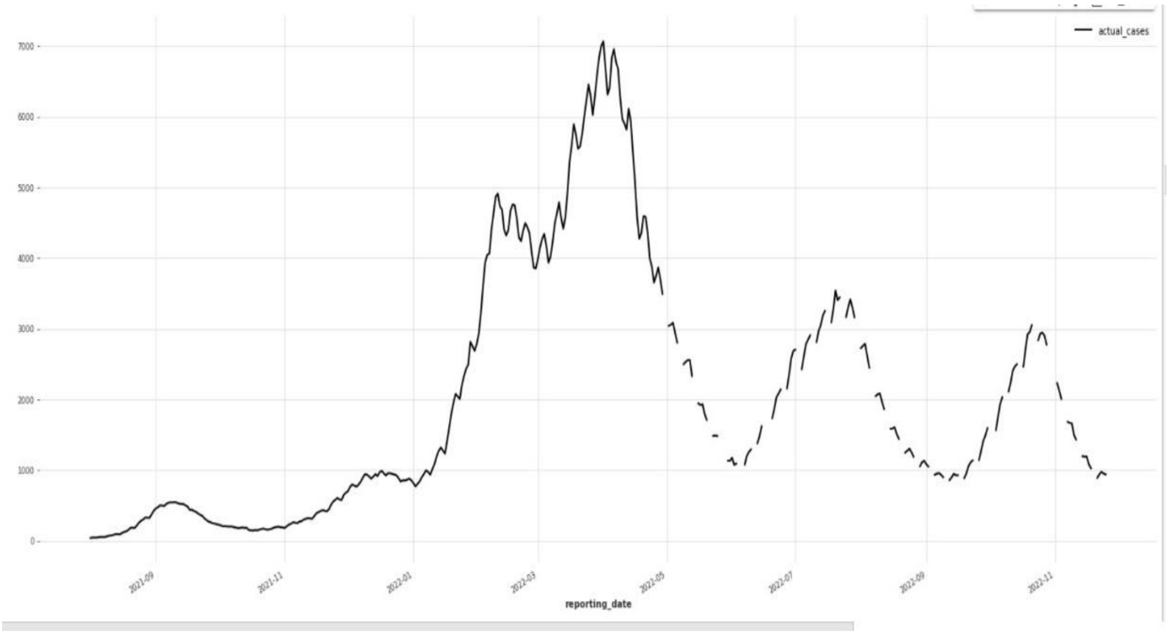


Figure 4 Missing values starting from May 2022

Schema Design

Author: Rahul Bhanushali

A database schema is an essential element of a relational database because it defines the structure and organization of the data, including the data types, attributes, and relationships between data entities. A well-designed schema helps to ensure that the data is stored efficiently and accurately, and it can help to avoid redundancy and improve the consistency of the data.

One of the main benefits of a database schema is that it helps to eliminate data redundancy, which is the unnecessary duplication of data in a database. By organizing the data into tables with clearly defined relationships, a schema can help to ensure that data is stored only once and is accessed consistently throughout the database. It can also help to improve the performance of the database.

Another important benefit of a database schema is that it helps to improve the consistency of the data. By defining the data types and constraints for each column in a table, a schema can help to ensure that the data is entered and stored in a standardized format. A database schema is an essential element of a relational database that helps to ensure the efficiency, accuracy, and consistency of the data.

Considering the Covid-19 data fetched from the above-mentioned source, a relational database schema was designed to store the data in the database. Four main tables were created in the schema to store data from the daily report, rate of hospitalization, rate of deceased, and rate of infection from the weekly report, respectively. Insert timestamp and update timestamp fields were added in all four tables to differentiate the data records for separate days/weeks. An index table was created for the age category to maintain consistency in case introducing a new age category in the source.

The main challenge faced during the schema design was to define the relationship between all tables using the foreign key and primary key constraints. This was because, in any of the four main tables, no single field can work as a primary key which in turn cannot be referred to as a foreign key in other tables. For instance, the table with weekly data on the rate of infection, timestamp or city name, or age category alone cannot be defined as the primary key due to duplicate values. Hence, a combination of fields was selected as primary in the main tables to identify the single record. So, in the table with weekly data on the rate of infection, a combination of values from fields insert timestamp, city name, and age category was defined as the primary key. Now, such primary keys were referred to as foreign keys in other tables in the database to establish a relationship between them. Schema design consisting of tables and their relationships can be easily understood with the help of an Entity Relationship (ER) diagram as shown below.

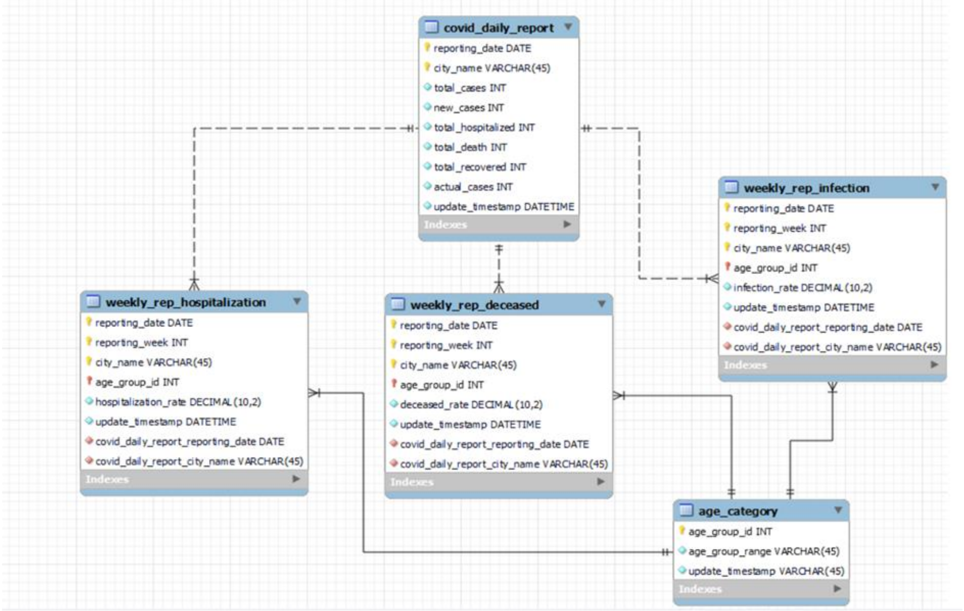

covid_daily_report
reporting_date DATE
city_name VARCHAR(45)
total_cases INT
new_cases INT
total_hospitalized INT
total_death INT
total_recovered INT
actual_cases INT
update_timestamp DATETIME
Indexes
weekly_rep_infection
reporting_date DATE
reporting_week INT
city_name VARCHAR(45)
age_group_id INT
infection_rate DECIMAL(10,2)
update_timestamp DATETIME
covid_daily_report_reporting_date DATE
covid_daily_report_city_name VARCHAR(45)
Indexes
weekly_rep_hospitalization
reporting_date DATE
reporting_week INT
city_name VARCHAR(45)
age_group_id INT
hospitalization_rate DECIMAL(10,2)
update_timestamp DATETIME
covid_daily_report_reporting_date DATE
covid_daily_report_city_name VARCHAR(45)
Indexes
weekly_rep_deceased
reporting_date DATE
reporting_week INT
city_name VARCHAR(45)
age_group_id INT
deceased_rate DECIMAL(10,2)
update_timestamp DATETIME
covid_daily_report_reporting_date DATE
covid_daily_report_city_name VARCHAR(45)
Indexes
age_category
age_group_id INT
age_group_range VARCHAR(45)
update_timestamp VARCHAR(45)
Indexes

Figure 5 Entity Relationship (ER) diagram

Data Storage

Author: Rahul Bhanushali (70%) and Rahul Chhabadiya (30%)

To store crawled Covid-19 data from Homepage Landesuntersuchungsamt (rlp.de) in a MySQL database using Python we must perform the following steps:

1. Install the necessary Python packages: Install the MySQL-connector-python package in order to connect to a MySQL database from Python.
2. Import the following modules: MySQL. Connector.
3. Connect to the database: Use the mysql.connector.connect() function to connect to the MySQL database. Provide database credentials, including the user ID and password, to establish the connection.
4. Create an INSERT statement: Use an INSERT statement to add the extracted data to the database. Make sure to include placeholders for the data that you will be inserting.
5. Execute the INSERT statement: Use the cursor.execute() method to execute the INSERT statement and insert the data into the database.
6. Commit the changes: Use the conn.commit() method to commit the changes to the database.
7. Close the connection: Use the conn.close() method to close the connection to the database.

Application Architecture

Author: Rahul Bhanushali

Pipeline

An ETL (Extract, Transform, Load) data pipeline is a process for extracting data from various sources, transforming it into a format suitable for analysis or other purposes, and loading it into a database or other storage system.

1. **Extract**: The first step in the ETL process is to extract the data from the source. In this case, the source Homepage Landesuntersuchungsamt (rlp.de) is a website and the data is being crawled using tools such as requests and BeautifulSoup in Python. These allow us to send HTTP requests to the website to extract the data files of interest.
2. **Transform**: Once the data has been extracted, it is typically transformed into a format that is more suitable for storage and analysis. This step may involve a variety of tasks, such as cleaning the data (e.g., removing missing or invalid values), normalizing the data (e.g., scaling numerical values), and formatting the data (e.g., converting strings to dates). The specific transformation steps you need to perform will depend on the structure and quality of the data, as well as the requirements of the database or other storage system you are using.
3. **Load**: The final step in the ETL process is to load the transformed data into the database or other storage system. This is typically done using a database connector such as mysql-connector-python, which allows you to connect to the database from Python and execute SQL statements to insert the data. Depending on the size and complexity of the data, you may need to use techniques such as batch inserts or parallel loading to optimize the performance of this step.

ETL Pipeline to perform the complete cycle of extraction, transformation, and storing of daily and weekly Covid-19 data respectively was created by following the above steps in python.

Automation

Author: Rahul Bhanushali

Schedulers are useful for automating tasks that need to be run regularly. By using a scheduler, one can specify when a task should be executed, and the scheduler will take care of running the task at the specified time. This can be especially useful in cases where it’s required to run a task regularly but avoid manually triggering it each time.

As our source data is updated on both a daily and weekly basis, it is essential to schedule all tasks of data storage rather than triggering the code manually every time. All tasks in the ETL pipeline were automated using the ‘schedule’ library in python to perform the extraction, transformation, and storing of daily and weekly Covid-19 data from the data source. By automating these tasks with a scheduler, we can save time and effort, and ensure that the tasks are being run consistently and reliably. Schedulers can also be configured to run tasks in parallel, which can further improve efficiency and reduce the time required to complete the tasks.

Logger

Author: Rahul Bhanushali

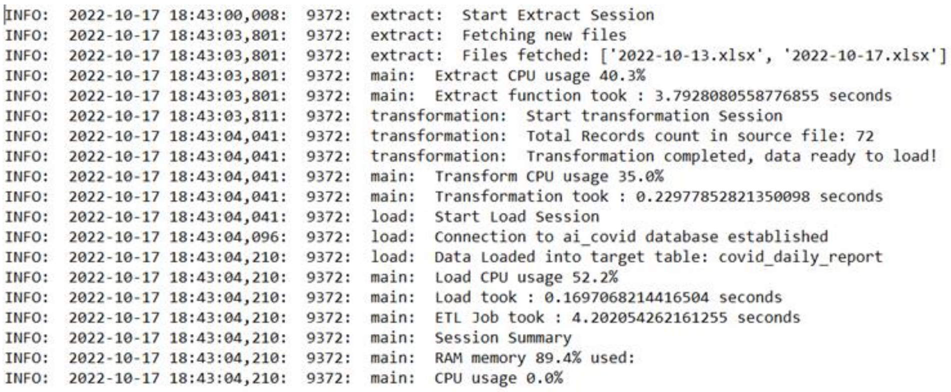
Loggers are important in ETL pipelines because they provide a way to track and monitor the execution of the pipeline. By adding logging statements to the code, we can capture notable events and messages that can help you understand what is happening during the pipeline's execution. This can be especially useful in cases where the pipeline is running for an extended period or where it is processing substantial amounts of data.

Some of the benefits of using a logger in an ETL pipeline include:

1. Debugging: Logging can help to identify and troubleshoot issues that arise during the execution of the pipeline. By examining the log messages, we can determine where problems are occurring and take corrective action.
2. Monitoring: Logging can be used to track the progress of the pipeline and monitor its performance. For example, we can log messages when certain tasks are completed, or when the pipeline encounters errors or other issues.
3. Auditing: Logging can be used to create an audit trail of the pipeline's execution. This can help understand what happened during a particular run of the pipeline, or for identifying trends or patterns in the data.

Hence, to achieve the above benefits, we integrated a logger in python throughout the ETL Pipeline code for daily and weekly data storage. This logger creates and stores two separate log files each for daily and weekly code containing log records of the execution of each line of code along with the timestamp. It also includes the records of exceptions and errors that occurred during the execution of pipeline code.

Below is the snippet of the log file created:



```
INFO:  2022-10-17 18:43:00,008:  9372:  extract:  Start Extract Session
INFO:  2022-10-17 18:43:03,801:  9372:  extract:  Fetching new files
INFO:  2022-10-17 18:43:03,801:  9372:  extract:  Files fetched: ['2022-10-13.xlsx', '2022-10-17.xlsx']
INFO:  2022-10-17 18:43:03,801:  9372:  main:  Extract CPU usage 40.3%
INFO:  2022-10-17 18:43:03,801:  9372:  main:  Extract function took : 3.7928080558776855 seconds
INFO:  2022-10-17 18:43:03,811:  9372:  transformation:  Start transformation Session
INFO:  2022-10-17 18:43:04,041:  9372:  transformation:  Total Records count in source file: 72
INFO:  2022-10-17 18:43:04,041:  9372:  transformation:  Transformation completed, data ready to load!
INFO:  2022-10-17 18:43:04,041:  9372:  main:  Transform CPU usage 35.0%
INFO:  2022-10-17 18:43:04,041:  9372:  main:  Transformation took : 0.22977852821350098 seconds
INFO:  2022-10-17 18:43:04,041:  9372:  load:  Start Load Session
INFO:  2022-10-17 18:43:04,096:  9372:  load:  Connection to ai_covid database established
INFO:  2022-10-17 18:43:04,210:  9372:  load:  Data Loaded into target table: covid_daily_report
INFO:  2022-10-17 18:43:04,210:  9372:  main:  Load CPU usage 52.2%
INFO:  2022-10-17 18:43:04,210:  9372:  main:  Load took : 0.1697068214416504 seconds
INFO:  2022-10-17 18:43:04,210:  9372:  main:  ETL Job took : 4.202054262161255 seconds
INFO:  2022-10-17 18:43:04,210:  9372:  main:  Session Summary
INFO:  2022-10-17 18:43:04,210:  9372:  main:  RAM memory 89.4% used:
INFO:  2022-10-17 18:43:04,210:  9372:  main:  CPU usage 0.0%
```

Figure 6 Execution log

ML model integration:

Authors: Martin Ebuka Okolie, Md Naiem Siddique and Boyon Dey Shipon

In this research project we implemented several different models to analyse the impacts of the COVID-19 pandemic in the Rhineland Palatinate region. These models included:

SEIRD: This is a mathematical model that is used to understand the spread and impact of infectious diseases. It stands for Susceptible, Exposed, Infectious, Recovered, and Dead, and is used to predict the number of people in each of these categories over time.

Random Forest: This is a machine learning algorithm that is used for classification and regression tasks. It works by creating an ensemble of decision trees and making predictions based on the majority vote of the trees.

ARIMA: This is a statistical model that is used to analyse and forecast time series data. It stands for Autoregressive Integrated Moving Average and is commonly used in fields such as economics and finance.

Exponential smoothing: This is a method for smoothing out time series data to make forecasts. It works by assigning higher weights to more recent data and lower weights to older data, resulting in a smoothed curve that is more responsive to latest trends.

Catboost: This is a machine learning library that is specifically designed for working with categorical data. It is a gradient boosting algorithm that can handle missing values and categorical features in the data.

Prophet: This is a library developed by Facebook for forecasting time series data. It is based on a decomposable time series model that can handle trends, seasonality, and other complex patterns in the data.

Polynomial regression: In polynomial regression, the relationship between the independent variable x and the dependent variable y is modelled by an nth degree polynomial in x.

Overall, these models provided a range of tools for analysing and understanding the impacts of the COVID-19 pandemic in the Rhineland Palatinate region.

The data for this research was retrieved from the LUA (Landesuntersuchungsamt) official website in Rhineland Palatinate using web scraping with a python script this retrieved excel files which were then processed and stored in an SQL database.

Machine Learning.

Model features and parameter selection

Author: Md Naiem Siddique

Correlation analysis

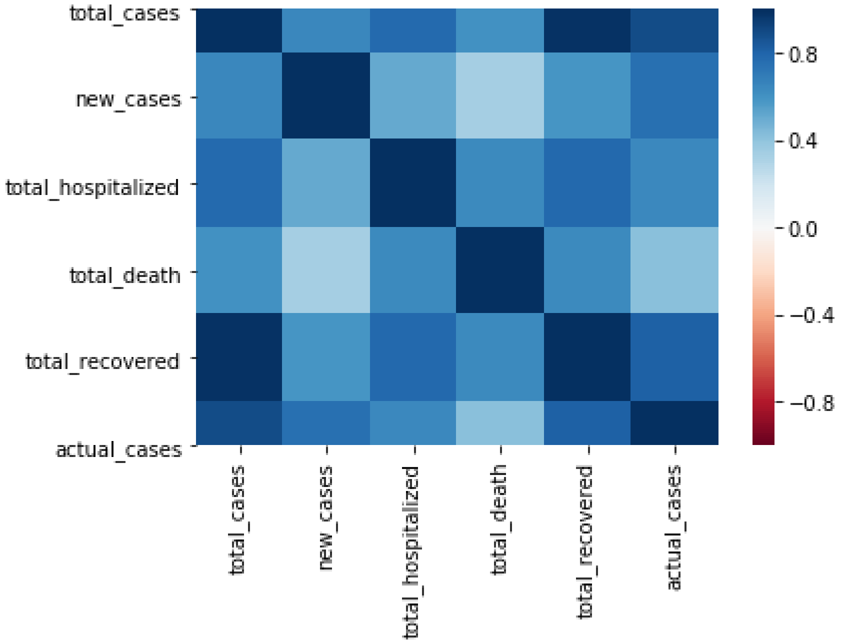


Figure 7 Correlation analysis from dataset

Parameter analysis

Daily incidence numbers (segregated by different districts of Rhineland Palatinate- RPL)

Total cases: Total reported cases from the beginning of pandemic (1 August 2021) which are confirmed by laboratory PCR tests.

(Daily) New cases: new infections in addition to the last date. It reflects the daily new cases except Monday because the new infection recorder in weekends (Saturday and Sunday) is added with Monday.

Hospitalization: Total number of hospitalization since the beginning (incremental)

Deceased: Total number of people reported dead because of and with covid infection.

Recovered: Total number of infected people who has been reported positive at least 21 calendar days before however neither deceased nor hospitalized (incremental).

Actual cases: Current actual cases of infected (reported at least 28 days before) and hospitalized but not deceased. Where, Actual = Total – Recovered – Deceased

Weekly normalized incidence rates (per 100000 population, (segregated by different districts of Rhineland Palatinate))

Age: The weekly (YYYYKWNN_COVID-19_Wochenbericht_RLP) incidence rate provided by LANDESUNTERSUCHUNGSAMT is categorized between different age groups i. less than 20 years ii. 20 – 59 years iii. 60 and 60+ years.

Overall rate: Infection percentage for each district (denoted as column “Rheinland-Pfalz”).

Overall rate+ (including US armed force): Infection percentage for each district (denoted as column “+USAF”) including the population of US armed forces deployed in that district.

Hospitalization: A national value is reported for the 7-day hospitalization incidence (starting form 24-11-2021) determined by the number of hospitalizations among SARS-CoV-2 cases. Hospitalization is admission to an acute care hospital due to COVID-19, of unknown or other known cause.

Vaccination

Age group: The vaccination and its effectiveness are categorized in several different age groups. In vaccination effectiveness measurement, 9 different age groups are used. However in vaccination breakthrough there are only 3 groups (12-17, 18-59, 60++).

Reporting Date: Publication date of the weekly report. Note on the same reporting date contains numbers which are the summary of several weeks (last 4/8 calendar weeks)

Infection Category: Type of infection category upon which the vaccination and effectiveness data are such as symptoms, hospitalized, ICU care, deceased etc. All these categories considered the COVID positive cases from a PCR test. The information is given as total numbers and percentage of people among them which are fully vaccinated and not fully or unvaccinated.

Model Selection

Author: Md Naiem Siddique

Our objective in this regard is to study and implement a set of machine learning algorithms to answer our second research question. We want to supervise the model so that the accuracy of our forecast system improves periodically. To measure the accuracy of our models, we store this forecast data in the database and validate against the actual numbers.

We found that supervised regressions and regression tree models fits best for this research. There is an autocorrelation between the values. For example, today’s numbers of infected, deceased or recovered has correlation with the previous day values. Therefore, we prioritize the Auto Regressive Moving Average (ARIMA) algorithm. Another traditional compartmental model in viral infections is SIR(Susceptible-Infected-Recovered) is also explored. There has been several research on this model and different variations of this basic model can be found in literature. (Syrowatka et al., 2021)from Boston Massachusetts published a meta-analysis and reviewed 183 articles where authors recommended several artificial intelligence and machine learning techniques to fit them in different use cases of covid infection analysis. They recommended augmenting traditional models such as neural networks and data-driven recurrent neural networks to forecast decease dynamics and effects of interventions. However, in our research project we kept neural network out of scope as it fits best with a large set of complex non-linear data. In our scenario, regression-based techniques are implemented for initial understanding of the decease spreading dynamics. Nevertheless, neural network would be vital when additional factors contributing to covid infection such as pharmaceutical and non-pharmaceutical interventions data is also considered. We have chosen rather simplistic model implementation such as univariate and multivariate modelling over the parameters number of infection, hospitalized, recovered, and deceased.

Selected models for further analysis

- ARIMA
- Prophet

- Cat-boost
- Exponential smoothing
- Random Forests
- SEIRD
- Polynomial regression

ARIMA
Author: Martin Ebuka Okolie

Model ARIMA Early in the 1970s, Box and Jenkins created the autoregressive integrated moving average (ARIMA) model, a time series prediction method that has been widely used to predict infectious diseases. A mathematical model called ARIMA uses past data to predict future values of a variable. The fundamental ARIMA equation is as follows:

$$\Theta_P(B^s)\theta_p(B)\ (1 - B^s)^D(1 - B)^d y_t = \phi_Q(Bs)\ {}^{\varphi}{}_q(B)\varepsilon_t$$

In this equation, y_t stands for the predictive value, B for the backward shift operator, t for time series residuals, and Θ_P, θ_p, ϕ_Q, and ${}^{\varphi}{}_q$ stand for the four ARIMA model parameters, p, q, P, and Q, respectively.

The degrees of the seasonal and trend differences are shown here by the letters d and D, respectively. The order of auto-regression, seasonal auto-regression lag, order of moving average, seasonal moving average, and seasonal periodicity are all represented by the ARIMA model parameters p, P, q, Q, and s, respectively.
The ARIMA model is generally referred to as ARIMA (p, d, q) (P, D, Q) s. Because the daily confirmed COVID-19 instances in the time series were non-seasonal data, the ARIMA model was stated in this study as ARIMA (p, d, q), and its equation is as follows:

$$\Theta_P\ (B)\ ((1 - B)^d y_t = {}^{\varphi}{}_q\ (B)\varepsilon_t$$

There are various processes involved in building the ARIMA model. To see if the time series was stationary, the daily verified COVID-19 case sequence was first plotted.

Using difference and log transformations, non-stationary time series sequences were converted into stationary sequences. Second, the graphs of the autocorrelation and partial auto-correlation function were analysed to estimate the parameters of the ARIMA model. After difference and log transformations, the parameters p, P, q, and Q were calculated using graphs of the auto-correlation function (ACF) and partial auto-correlation function (PACF). The potential ARIMA model was initially identified. Third, the Ljung-Box (Q) test and the t-test were used, respectively, to diagnose and evaluate the ARIMA model. The daily COVID-19 case time series' residuals must be white noise for the Ljung-Box (Q) test to be significant ($p > 0.05$ significant level). The significance of the parameters of each proposed ARIMA model was assessed using a t-test.

The minimal normalized BIC, RMSE, and maximum R-square values determine the best model, while the residuals are white noise sequences.
In time series forecasting, the Bayesian information criterion (BIC) is frequently employed for model selection. It was created by Schwarz, and its definition is given as follows:

$$\text{BIC} = -2 \ln(L) + \ln(n)0 * k$$

where L is the model's likelihood function's maximal value, n is the sample size, and k is the number of parameters the model estimates. To validate the model's suitability, the normalized Bayesian information criterion (BIC) was applied. The better the fit of the model, the smaller the value of the normalized BIC. (Zhao et al., 2022)

Prophet model

Author: Martin Ebuka Okolie

The Prophet model, an open-source time-series forecasting algorithm, was created by Facebook in 2017, and can be run using R or Python. The basic formula for the Prophet model is as follows:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Here, *y(t)* is the predictive value, *g(t)* is the trend function that models non-periodic changes in the time series of daily confirmed COVID-19 cases, *s (t)* signifies periodic changes (weekly characteristics of confirmed COVID-19 cases time series), and *h (t)* signifies the effects of holidays on potentially irregular schedules. For example, Christmas Day. εt signifies idiosyncratic changes that are not accommodated by the model.
In trend model *g (t),* there are two types of models: a saturating growth model and a one-piece linear model that covers numerous Facebook applications. The formula for the nonlinear saturation growth model is as follows:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))}$$

where *C* is the carrying capacity, *k* is the growth rate, and *m* is the offset parameter.
The formula for the piecewise logistic growth model is as follows:

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \alpha(t)T\delta)(t - (m + \alpha(t)T\gamma)))}$$

where δ is a vector of rate adjustments and γ is the correct adjustment at the change point.
The seasonality *s(t)* depends on the Fourier series to provide a viable model for periodic effects. This formula is expressed as follows:

$$s(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\frac{2\pi nt}{P} + b_n \sin\frac{2\pi nt}{P} \right)$$

where *a* is standard Fourier series, *P* is the periodic changes.
Holidays and events *h(t)* have a greater influence on predicting time-series performance because they do not follow a periodic pattern.

$$Z(t) = [1(t \in D1, \ldots 1(t \in DL)]$$

$$h(t) = Z(t)k$$

where *t* is during holiday *i* and *ki* is the holiday parameter and a prior *k* ~ normal (0, v2). (Zhao et al., 2022)

Catboost

Author: Martin Ebuka Okolie

The Catboost algorithm is an ordered boosting technique that focuses on resolving the boosting algorithm's overfitting problem. The ordered boosting approach constructs the model for computing the residual error using restricted data and then runs the model on the whole data set. Furthermore, including random permutation in the ordered boosting successfully minimises overfitting. The catboost algorithm creates a sample mean for variables in the same category that go through random permutation to cess categorical data. The Catboost technique increases training speed by combining feature combinations that aggregate variables from other ensemble algorithms such as random forest and gradient boosting. GridSearch or RandomisedSearch should be used to discover the ideal hyperparameter; catboost does not go through those phases as the default setting of the parameter is already satisfied. (Justin Shinjae Kim et al., 2021)

Exponential Smoothing

Author: Martin Ebuka Okolie

Exponential smoothing is a forecasting strategy that gives fresh observations a higher weight than older observations. The weights of previous observations are allocated in an exponentially decreasing order. It has been demonstrated that this weight distribution procedure outperforms the classic moving average methodology. Because moving average forecasting models provide the same weights to all data, this is the case. A weighting scheme based on time closeness delivers improved smoothed time series forecasting. There are three types of exponential smoothing models: single, double, and triple.
The mathematical expression for a single exponential smoothing is.

$$S_t = \alpha y_{t-1} + (1-\alpha)S_{t-1}, \quad 0 < \alpha \leq 1 \;\; t \geq 3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(1)$$

Where S_t is the smoothed value at time t, α is the smoothing parameter and y is the original observation.
Using equation 1, S_{t-1} can be represented as;

$$S_{t-1} = \alpha y_{t-2} + (1-\alpha)\, S_{t-2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Substituting equation 2 into equations 1

$$S_t = \alpha y_{t-1} + (1-\alpha)\, S_{t-2}\, [\alpha y_{t-2} + (1-\alpha)\, S_{t-2}] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(3)$$

Further simplification.

$$S_t = \alpha y_{t-1} + (1-\alpha)\, y_{t-2} + (1-\alpha)^2\, S_{t-2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Substituting for S_{t-2}, S_{t-3}................ S_2 is a series which can be expressed as:

$$S_t = \alpha \sum_{i=1}^{t-2} (1-\alpha)^{i-1}\, y_{t-1} + (1-\alpha)^{t-2}\, S_2\,,\; t \geq 2. \ldots\ldots\ldots\ldots\ldots\ldots\ldots.(5)$$

Thus, a smoothed value S_5 can be expressed as:

$$S_5 = \alpha\, [(1-\alpha)^0\, y_{5-1} + (1-\alpha)^1\, y_{5-2} + (1-\alpha)^2\, y_{5-3}] + (1-\alpha)^i \ldots\ldots\ldots\ldots\ldots..\; (6)$$

It can be demonstrated using geometric series that the weights $(1-\alpha)^i$ form a geometrically diminishing pattern with a total of unity.

In equations 1–6, we demonstrated that prior observations are allocated exponentially decreasing weights. This is where single exponential smoothing forecasting methods outperform moving average forecasting techniques.

Single exponential smoothing is superior to typical moving averages, but it does not account for trend. This is a significant limitation of its predicting accuracy.

As a result, the double exponential smoothing introduces a new constant. A mathematical expression for double exponential smoothing is

$$S_t = \alpha y_t + (1-\alpha)(S_{t-1} + b_{t-1}),\; 0 \leq \alpha \leq 1 \ldots\ldots\ldots\; (7)$$

$$S_t = \gamma(S_t - S_{t-1}) + (1-\gamma)\, b_{t-1},\;\; 0 \leq \gamma \leq 1 \ldots\ldots\ldots\; (8)$$

Equation 7 of the double exponential smoothing demonstrates that lag is eliminated by adjusting the smoothed value S_{t-1} with the trend b_{t-1}. The trend b_t has been updated in Equation 8.

Datasets with seasonality cannot be considered by the double exponential smoothing equation. Consequently, a triple exponential smoothing is required. The Holt-Winters approach is this. Mathematically, the Holt-Winters approach is stated as:

$$S_t = \alpha \frac{yt}{It-l} + (1-\alpha)\, (S_{t-1} + b_{t-1}) \ldots\ldots\ldots.(9)$$

$$S_t = \gamma\, (S_t - S_{t-1}) + (1-\gamma) b_{t-1} \ldots\ldots\ldots\ldots\ldots..(10)$$

$$I_t = \beta \frac{yt}{St} + (1-\beta) I_{t-L} \ldots\ldots\ldots\ldots\ldots\ldots\ldots.(11)$$

$$F_{t+m} = (S_t + mb_t)\, I_{t-L+m} \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(12)$$

The Holt-Winters exponential smoothing forecasting models are dependent on both the trend and seasonality, as demonstrated in equations 9 to 12. These equations use the variables y, S, and B to denote the observation, smoothed observations, and trend factors, respectively. I stands for the seasonal index, F for the forecast m periods in the future, and t for the current moment.(Oladunni et al., 2021)

Random Forests

Author: Martin Ebuka Okolie

Each tree in a random forest depends on the values of a random vector that was sampled randomly and with the same distribution for all the trees in the forest. As the number of trees in a forest increases, the generalization error converges as to a limit. The strength of each individual tree in the forest and the correlation between them determine the generalization error of a forest of tree classifiers. Each node is divided using a random selection of features, producing error rates that are comparable to Adaboost but are noise-resistant more robust. Internal estimates keep track of inaccuracy, strength, and correlation; they are used to demonstrate how the splitting process responds to an increase in the number of features. Internal estimations are another method for gauging variable significance. Regression can also use these concepts.(Breiman, 2001)

SEIRD Model

Author: Md Naiem Siddique

SEIRD is the abbreviation of Susceptible-Exposed-Infected-Recovered-Deceased. This is a compartmental model typically associated with virology. It is an extension of the basic Susceptible-Infected-Recovered mathematical model which is widely used in modelling of infectious diseases such as influenza, Ebola, tuberculosis etc. Some other variations of base SIR models are

- eSIR (Wangping et al., 2020): eSIR is an extended Susceptible-Infected-Recovered model which also includes different intervention measures into the basic model. Phase adjusted preventive measures and time varying parameters are considered in this model extension.
- SEIR(Wu et al., 2020): Susceptible-Exposed-Infected-Recovered model which includes exposed population with the basic SIR model. Exposed are the susceptible population who had contact with the infected persons or lives in a close proximity with them. Travel, mobility and contact tracing information are generally used in identifying number of exposed people.
- SIRD(Kermack & Mckendrick, n.d.): This model includes deceased population with the base model. Since the deaths caused due to the infection has significant importance, it should be incorporated in the model to have a broader understanding of the disease. One of the core objectives of these epidemiological researches is to minimize the casualties. Therefore, this model provides an additional compartment to extend the model.
- A-SIRV(Marinov & Marinova, 2022): A-SIRV denotes “Adaptive SIR with vaccination” information. This is a time dependent model which is taking the vaccination information into account. A-SIRV tries to solve inverse problem of unknown time dependent rates and functions which is present in the publicly available COVID-19 dataset. There is also possibility to extend this model to understand the effects of non-pharmaceuticals interventions.

The main challenge of these models is selecting the right one and estimating the model parameters. There could be different approaches to estimate the parameters. Aspects such as geolocation, social connectedness, interventions, seasonality etc. has significant influence over the model parameters. Therefore, it is important to address these factors when estimating the parameters. A large dataset can be very useful to accurate estimation with the help of neural network and/or likelihood estimation and Bayesian inference.

Polynomial Regression

Author: Boyon Dey Shipon

Polynomial regression is one of the machine learning algorithms used for making predictions. Polynomial regression is a kind of regression analysis in those the relationship between the independent variables and the dependent variables is analyzed by a polynomial of the nth polynomial degree.
A polynomial regression model has the following form:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_h X^h + \varepsilon$$

Current trend analysis

Author: Martin Ebuka Okolie

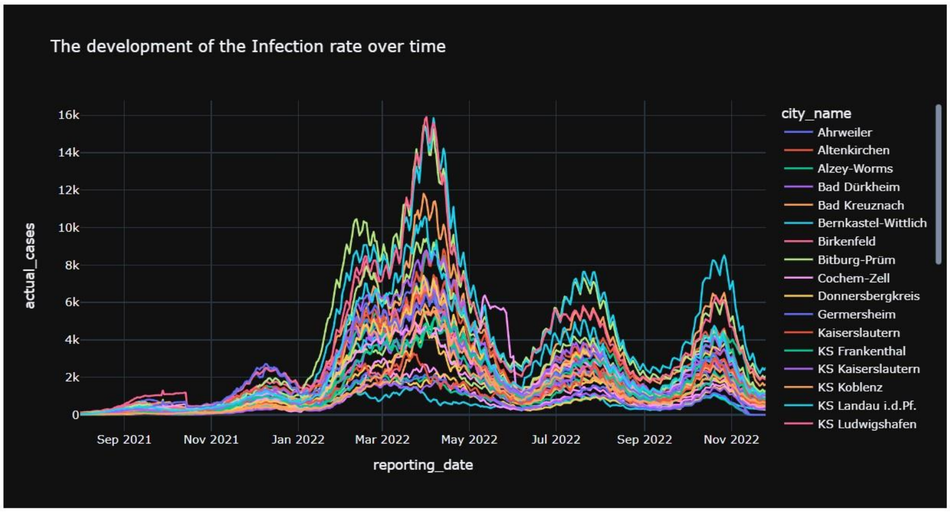


Figure 8 The trend of infection rates in Rhineland Palatinate's various regions over a year and two months.

The trend of infection rates in Rhineland Palatinate's various regions over a year and two months is depicted in the figure above. During this period, there were spikes in infection rates in April, August, and November. The spike and peak in March and April respectively were caused by the relaxation of the COVID-19 protection measures in March 2022, as reported by the Deutshse Welle.

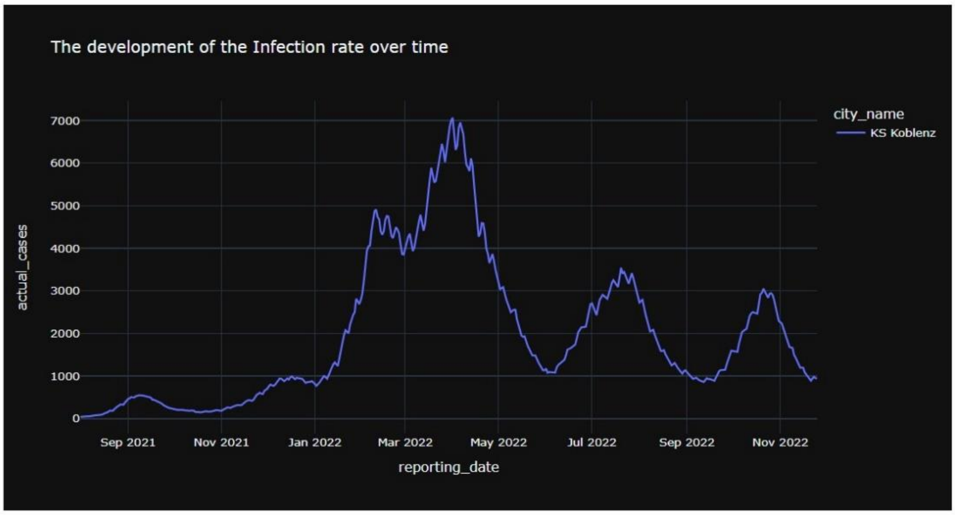


Figure 9 The trend of infection rates of in Koblenz over a year and two months.

The figure above shows the trend of infection rates of in Koblenz over a year and two months as we can observes that the pattern is similar to the other regions in Germany.

NOTE: The data has been manipulated at this point because there were dates with missing values from the original data set (Saturday and Sundays had no recorded data) so the missing values (NaN) were filled in using and interpolate function that takes the previous in the dataset in order to make the graph readable.

During the pre-processing we also tried to smoothen the dataset as smoothed values makes it easier for the algorithm to function. Although the results gotten from prediction with pre-processed smoothing values may be off, but it helps us identify patterns in the dataset that we might not see clearly with the dataset with unsmoothing values.

After applying a rolling window of 7 days (smoothing by finding the mean of the first seven days and assigning the mean to all the data points) to the dataset, here is what it looks like.

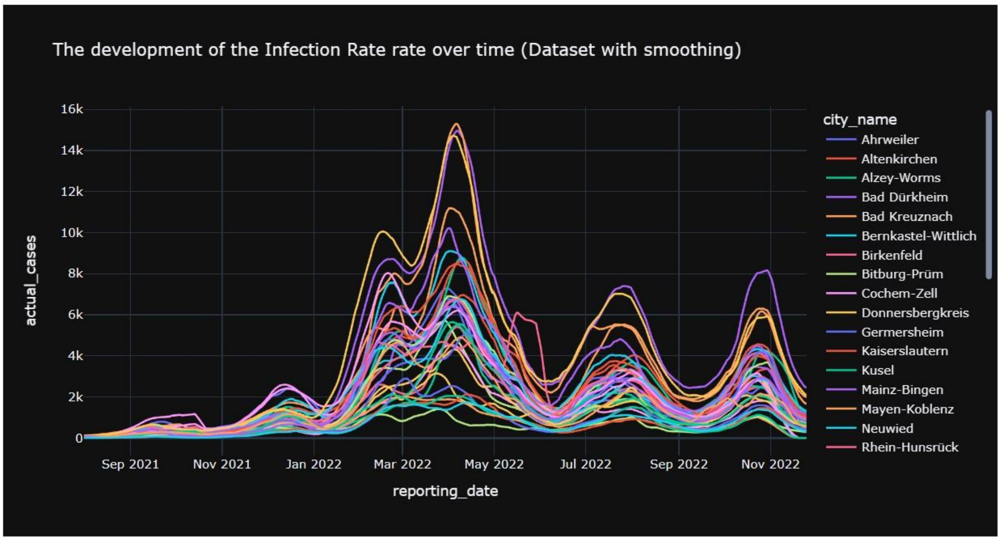


Figure 10 The effect of the 7-day rolling window smoothing method on the dataset

Above we can see the effect of the 7-day rolling window smoothing method on the dataset,

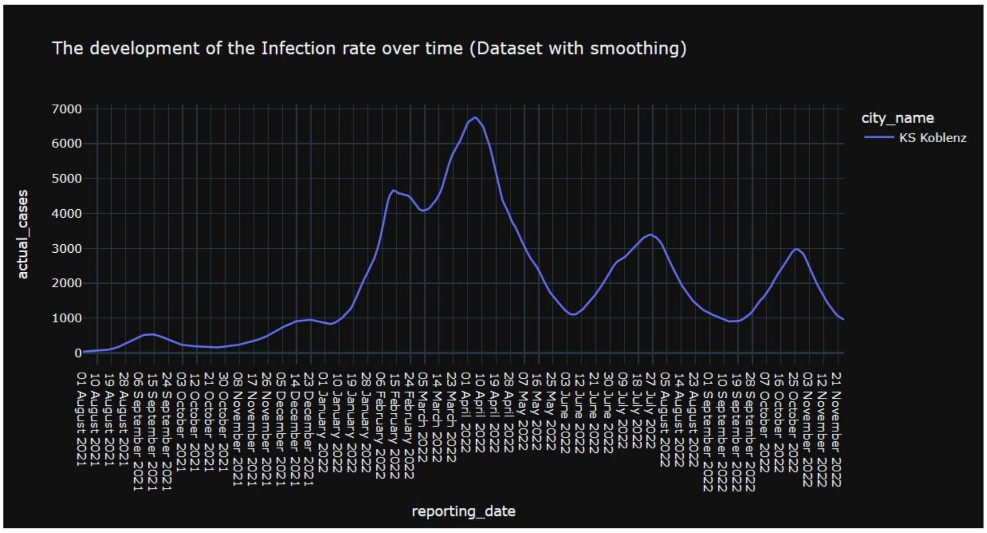


Figure 11 The effect of the 7-day rolling window smoothing method on the dataset (Single example, KS Koblenz)

Above we can see the effect of the 7-day rolling window smoothing method on the dataset (Single example, KS Koblenz)

Model Implementation

Darts

Author: Martin Ebuka Okolie

Darts is a Python module for intuitive time series forecasting and anomaly detection. It includes a range of models, including traditional ones like ARIMA and deep neural networks. The fit() and predict() functions, which are akin to those in scikit-learn, can be used to use each of the forecasting models in the same way. Additionally, the library makes it simple to backtest models, aggregate the predictions of other models, and take into consideration outside data. Univariate and multivariate time series and models are supported by darts. Some of the ML-based models provide rich support for probabilistic forecasting, and they may be trained on potentially enormous datasets with multiple time series.

Darts also offers extensive anomaly detection capabilities. For instance, it is trivial to apply PyOD models on time series to obtain anomaly scores, or to wrap any of Darts forecasting or filtering models to obtain fully fledged anomaly detection models.

In this research DARTS was used in the python script to build the different machine learning models since we were working with more than one. It helped to concurrently train, test and evaluate each model all in one place. This made the process efficient and effective. Google collab was used as the IDE for the writing the code contained in the python script because it enables the usage of the DARTS module.

NOTE: All predictions were executed as using Univariate timeseries. (Only the target value column was used for prediction in the different ML algorithms, ie infection rate, hospitalization rate and deceased rate)

The *future_periods* in this paper and the coding section of this research represents how much time into the future our models forecasted. In this research the *future_periods* used for forecasting was 14 days.

The data was split in to training and testing sets, 95% for training and 5% for testing before being fitted with the different algorithms. Below we can see the split in the values for the KS Koblenz region. (Smoothing values is displayed below and it is the same for the dataset without smoothing values)

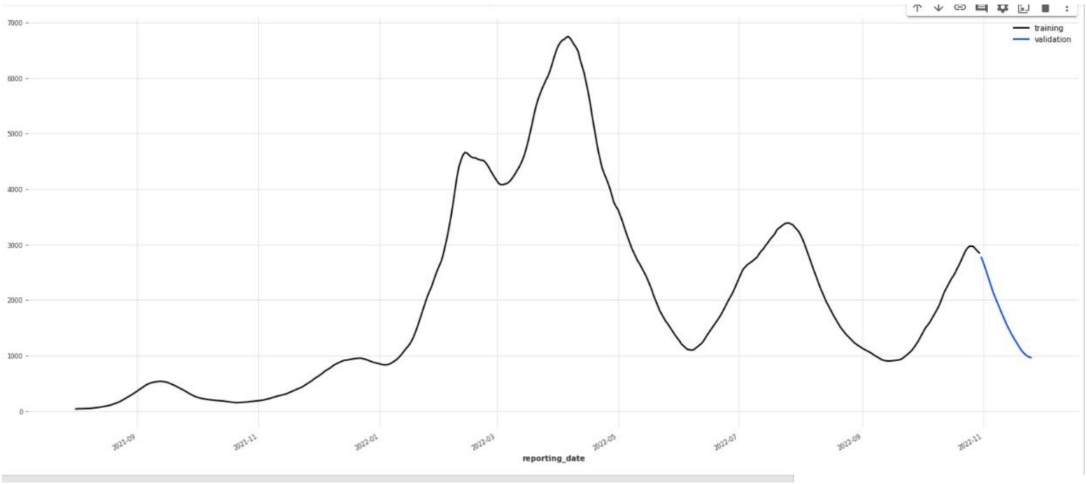


Figure 12 Splitting the dataset into training and testing 95% and 5% respectively.

After feeding the model with the different data sets and fitting with the different algorithms, (with and without smoothing values) we obtained results, and they are displayed in the figures below.

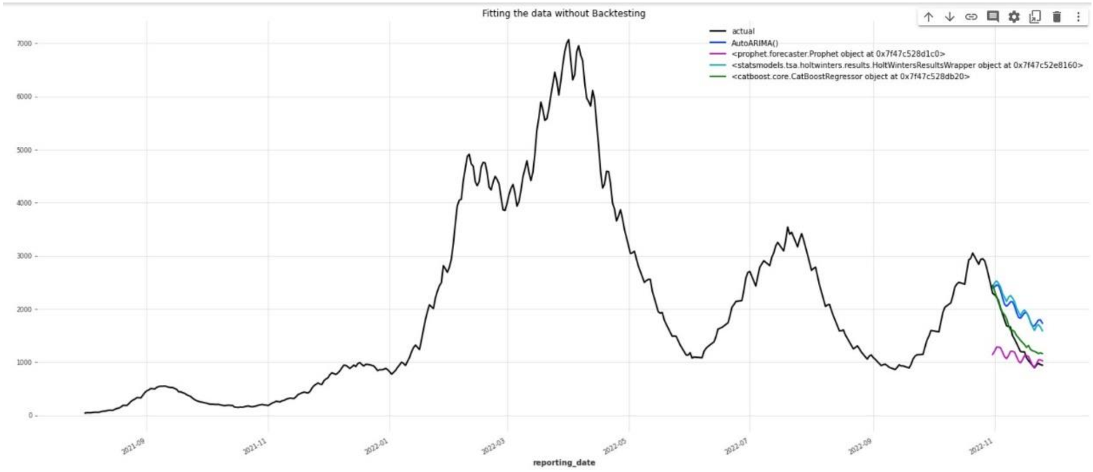


Figure 13 Graph after fitting the different models.

With smoothing values

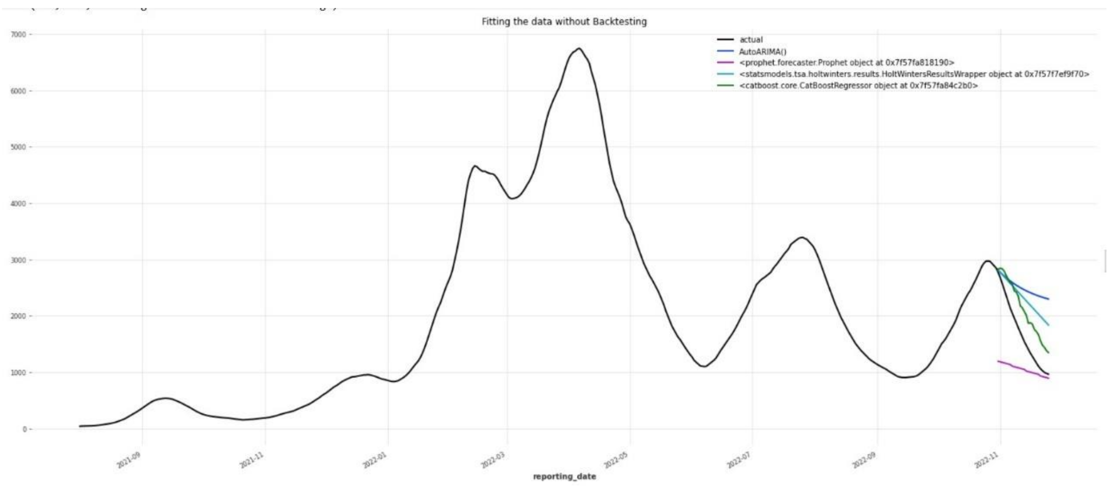


Figure 14 Graph after fitting the different models (Dataset with smoothing)

Backtesting is the process of simulating previous forecasts.

Backtesting is more robust than splitting data into train/validation/test sets since it allows us to examine the model's performance if we had used it previously by replicating predictions that would have been produced. It may take some time to generate because the model is (by default) re-trained each time the simulated forecast period advances.

Such simulated forecasts are always stated in terms of a forecast horizon, which is the number of time steps between the prediction time and the forecast time.

By doing back tests we are adding layer of confidence in the results we would obtain when we make the forecasts.

In the figures below, we can observe the results obtained from running backtests on the fitted models.

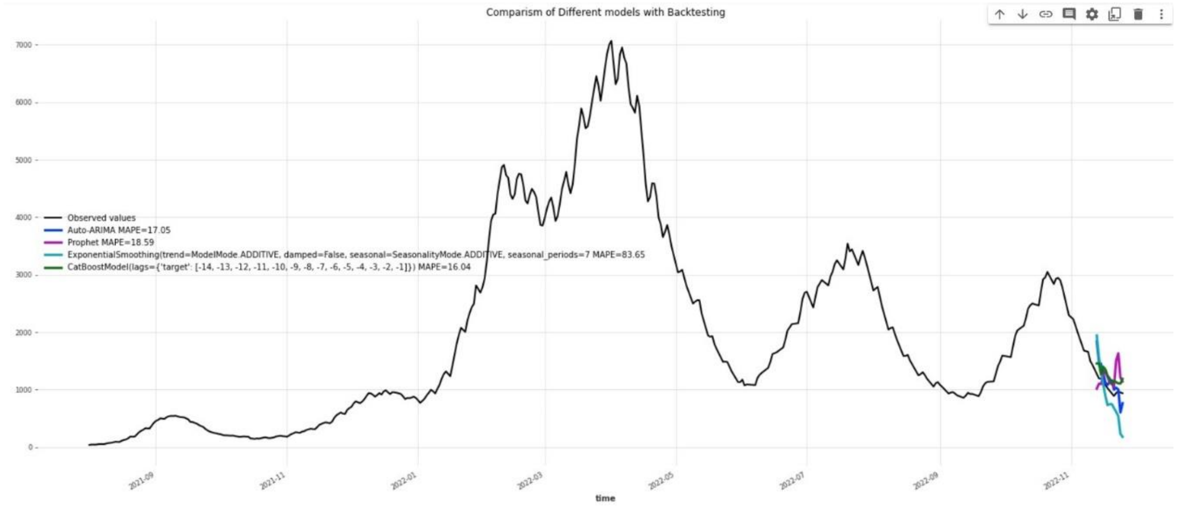


Figure 15 The results obtained from running backtests on the fitted models.

Applying backtesting to the dataset with smoothing values

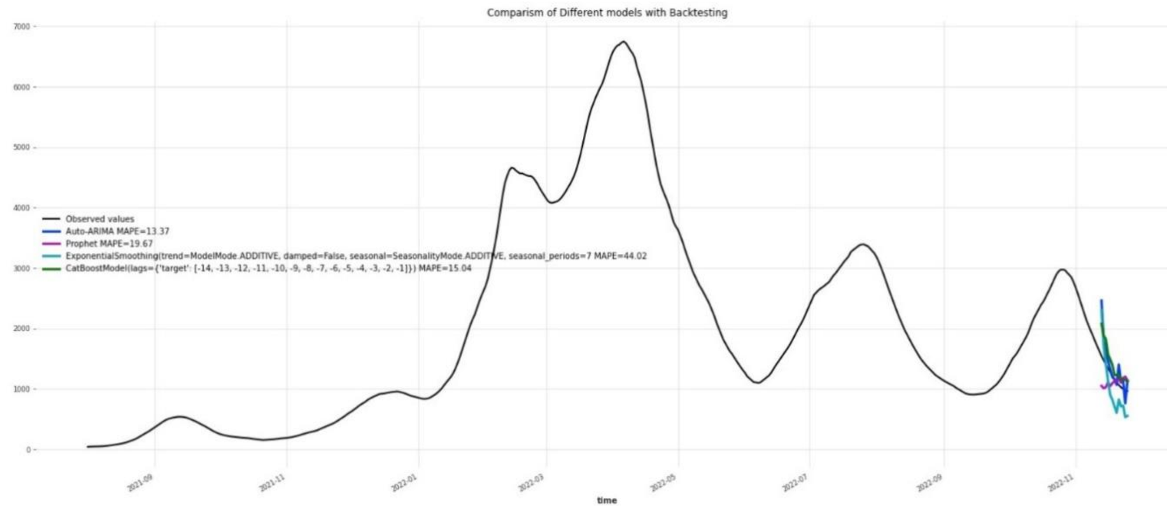


Figure 16 Results obtained from applying backtesting to the dataset with smoothing values.

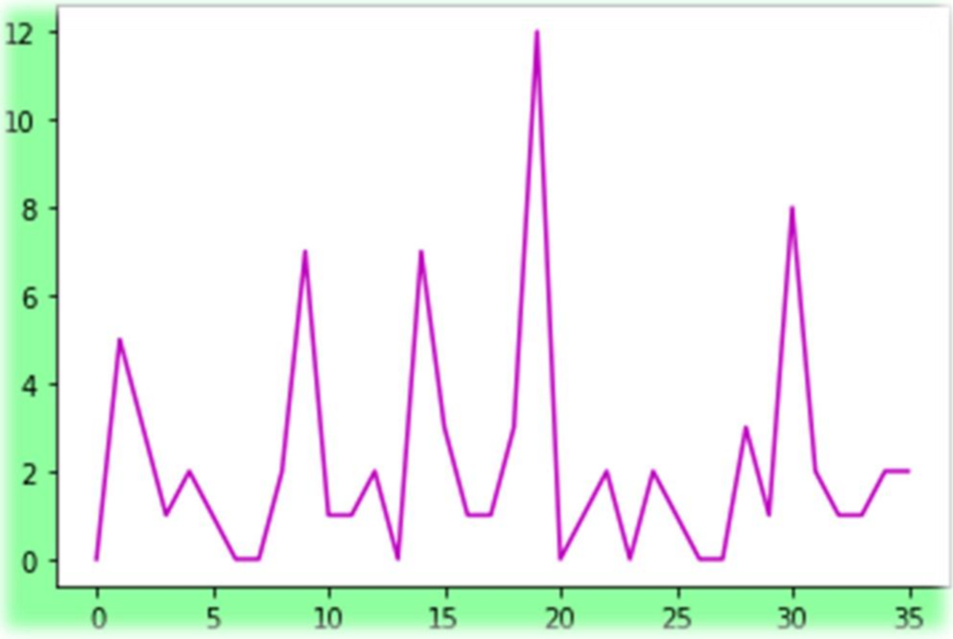
After the models were concurrently fitted using DARTS, the model was then ready to be created and that was done thereafter, in the next section we are going to show the method of model evaluation use in the research work.

NOTE: We chose the daily data set over the weekly data set since the models performed better when trained on the daily dataset. This might be due to the daily dataset having more datapoints than the weekly dataset.

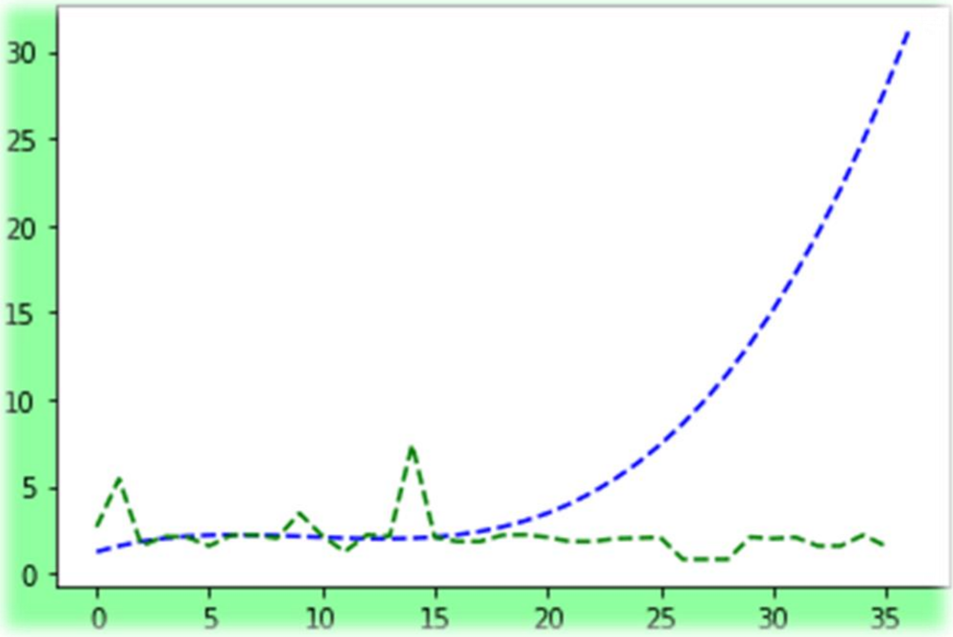
Polynomial regression

Author: Boyon Dey Shipon

When there is a nonlinear relationship between the predictor variable(s) and the response variable, we can fit a regression model using the polynomial regression method. I initially created the data using a few columns from the weekly Covid data. Although a curving line was intended, the data was not plotted as expected, resulting in a plot similar to the one shown below.



This resulted in two lines where the best fit line was not fitting at all because the outcome was not what was expected during data preparation. The outcome is displayed below.



Model Evaluation

Author: Boyon Dey Shipon

The issue that arose for not fitting with the best fit line is that in order to implement polynomial regression as a prediction algorithm, we must rely on the past and upcoming weeks' amount of data, both of which are always changing.

Authors: Martin Ebuka Okolie and Naiem Siddique

The different models were evaluated and compared using evaluation metrics like RMSE (Root Mean Square Error) and MAPE (Mean Average Percentage Error).

MAPE is a popular tool for determining prediction accuracy. It also calculates the forecast's percentage inaccuracy in reference to the actual data. It does not distinguish between them since it determines the average error across time or across goods. This implies that it makes no assumptions about which day or product to forecast better. It is computed as follows:

$$MAPE = \frac{1}{n} \sum_{t-1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

RMSE is a commonly used measure of the discrepancies between predicted and observed values (sample or population values) by a model or estimator. The RMSE is defined as the square root of the second sample moment of discrepancies between forecasted and observed values, or the quadratic mean of these differences. When calculated across the data sample used for estimate, these deviations are termed residuals, and when computed out-of-sample, they are called errors (or forecast errors). The RMSE aggregates the magnitudes of prediction errors for numerous data points into a single measure of predictive capacity. RMSE is a measure of accuracy used to evaluate forecasting errors of various models for a specific dataset rather than between datasets.

$$\text{RMSE}= \sqrt{\frac{\sum_{i-1}^{N}(actual-estimated)^2}{No\ of\ non-missing\ datapoints}}$$

In the table below, the evaluation metric values obtained from each model for the city of Koblenz after fitting with Dataset with smoothed values and Unsmoothed values is shown.

Evaluation metrics	ARIMA	Catboost	Exponential smoothing	Prophet	SEIRD model	Random forests
MAPE without smoothing	48.170121	13.291044	49.199338	20.960808	38.703645	NA
MAPE with smoothing	63.070561	30.897225	48.073879	32.481849	46.954863	NA
RMSE without smoothing	609.910264	173.769997	625.407407	513.620789	552.48789	396.922129
RMSE with smoothing	916.016300	462.968307	698.504012	797.326631	673.41289	404.37651

Figure 17 MAPE and RMSE values of each model with and without smoothing dataset.

The evaluation metrics for all the other cities can be obtained by running the python script for the machine leaning model.

From the table above we can determine that the Catboost model performs the best as it has the lowest metric values compared to the other models we implemented.

Visualization

Author: Pathey Atulkumar Pandya

A visualization is a crucial tool in data analysis and communication, as it can help make complex data more easily understandable and accessible. By creating visual representations of data, patterns, trends, and relationships can be more easily identified and understood, making it easier to make decisions and communicate insights to others. Visualization can also help to identify outliers, errors, or other issues in the data that might not be apparent from looking at the raw numbers alone. Additionally, visualization can also be used in data storytelling which is a technique to communicate complex data to a wide audience engagingly and effectively.

There are many different visualization tools available, each with its strengths and weaknesses. Some of the most popular visualization tools are Tableau, Microsoft Power BI, QlikView, Looker, Microsoft Excel, etc. These are some examples, there are many other visualization tools out there that are designed for specific use cases and industries. For our research, we were choosing Tableau.

Tableau is a widely used and popular data visualization tool that offers a variety of unique features that set it apart from other visualization tools:

Wide range of data connectors: Tableau offers a wide range of data connectors that allow users to connect to a variety of data sources, including databases, spreadsheets, and cloud-based services.

Data blending: Tableau allows users to combine data from multiple sources into a single, unified view, making it easy to analyze and compare data from diverse sources.

Advanced visualizations: Tableau offers a wide variety of visualization types, including bar charts, line charts, scatter plots, maps, heat maps, and more.

Dynamic dashboards: Tableau allows users to create dynamic dashboards that update in real time as the data changes.

Data exploration and analysis: Tableau provides a variety of tools for filtering and drilling into data, making it easy to explore and analyze data to find insights.

Mobile Compatibility: Tableau allows you to create visualizations that are accessible on different devices including mobile phones, tablets, and computers.

Advanced Calculation & Analytics: Tableau provides advanced data analytics and calculations, including clustering, forecasting, and trend analysis which can be easily performed in the tool.

We were using all these features while making the visualization from our data. (Source)

Tableau – MySQL connection

Author: Rahul Chhabadiya

Tableau is a data visualization tool that allows users to connect to various data sources, including MySQL. To connect Tableau to a MySQL database, you'll need to have the MySQL Connector/ODBC driver installed on your computer. Once the driver is installed, you can connect to the database by going to the "Connect" menu in Tableau and selecting "MySQL" as the data source.

Enter the necessary connection information, such as the server name “localhost”, port “3306”, and database name “ai_covid”, as well as MySQL username “root” and password “root”. Once connected, you'll be able to see the tables in your MySQL database and drag them into the "Sheet" area to start creating visualizations.

Dashboard

Author: Rahul Chhabadiya (40%) and Pathey Atulkumar Pandya (60%)

In Visualization, there are two dashboards, one is for historic data Visualization with MySQL, and the other one is for Machine Learning and Forecasting with excel sheets.

- Historic dashboard – Historic Dashboard includes a Map view, Line Graph, and Comparison View (based on weekly data)

In Map view, the city with the highest number of total covid cases has the biggest color bubble. The latitude and longitude used to pinpoint all the cities of Rheinland-Pfalz. There is an after-click event where the user can select more than two cities to open a comparison view which also includes a date filter for a better user experience.

On the top of the dashboard, the text box contains the total number of cases, deceased, and recovered which in this case are from the 1st of August 2021. These numbers are generated with the help of Tableau's calculated field. This number updates automatically when there would be an update in the database.

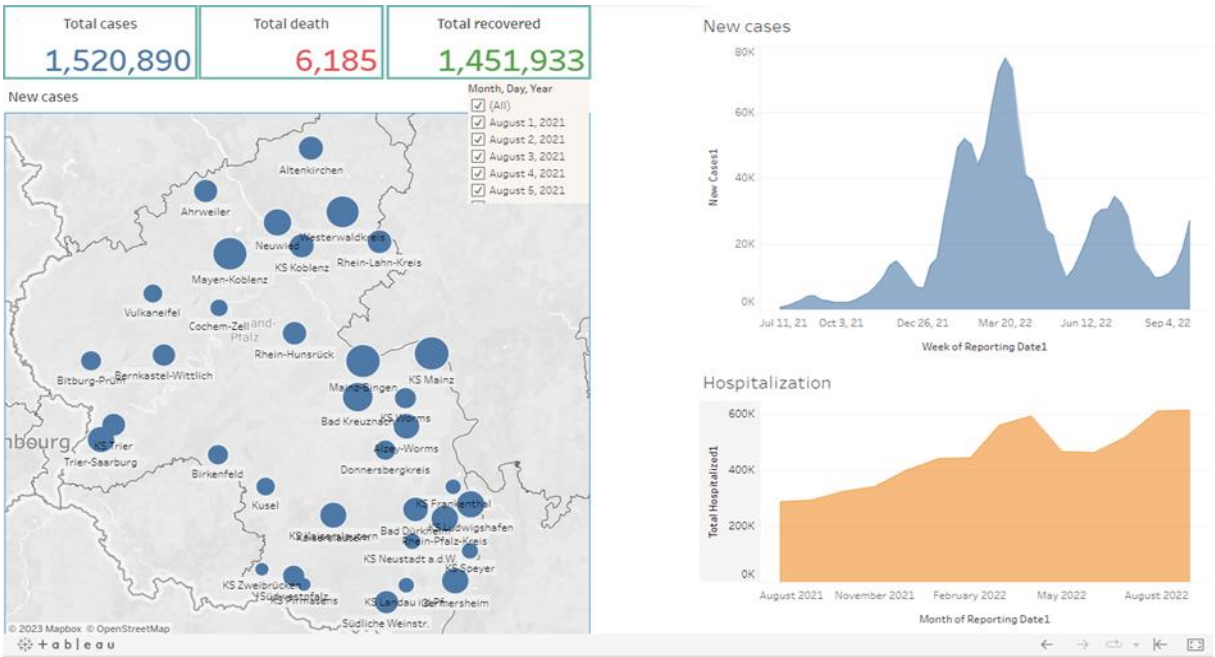


Figure 18 Dashboard for historic data

After selecting two or more cities from the map view the below view would be displayed.

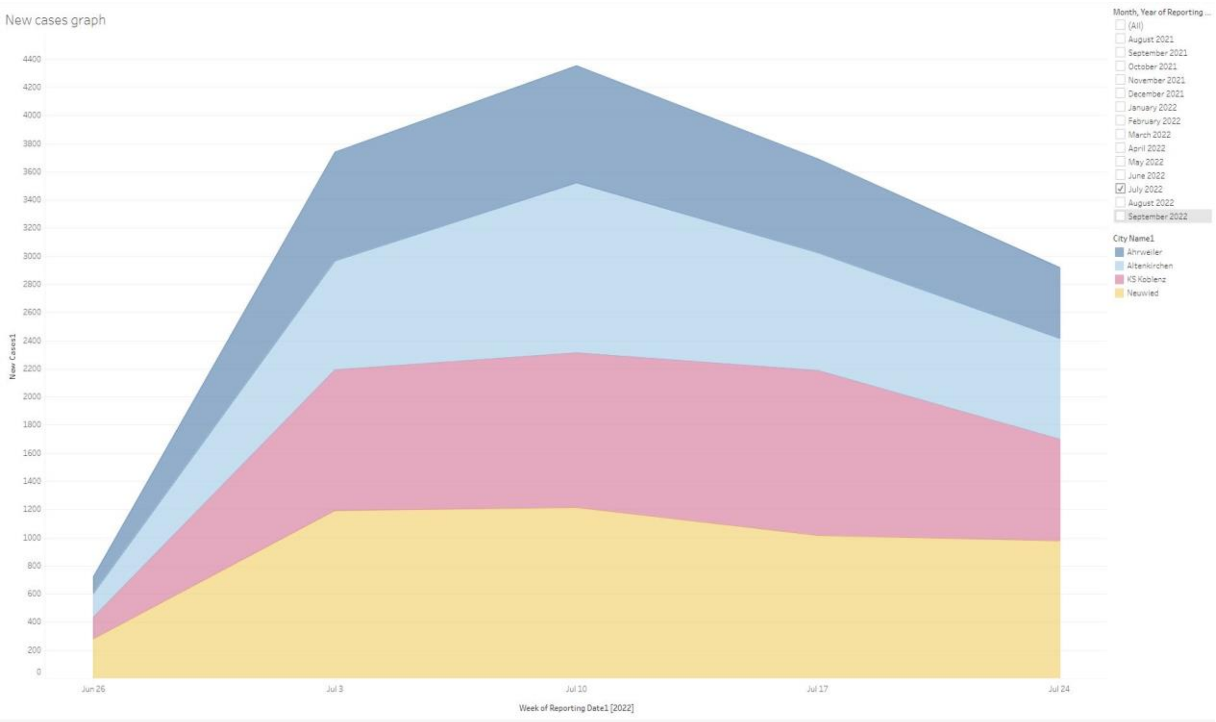


Figure 19 Comparison view of multiple locations

Prediction dashboard – Predicted data generated by ML model (based on weekly data)

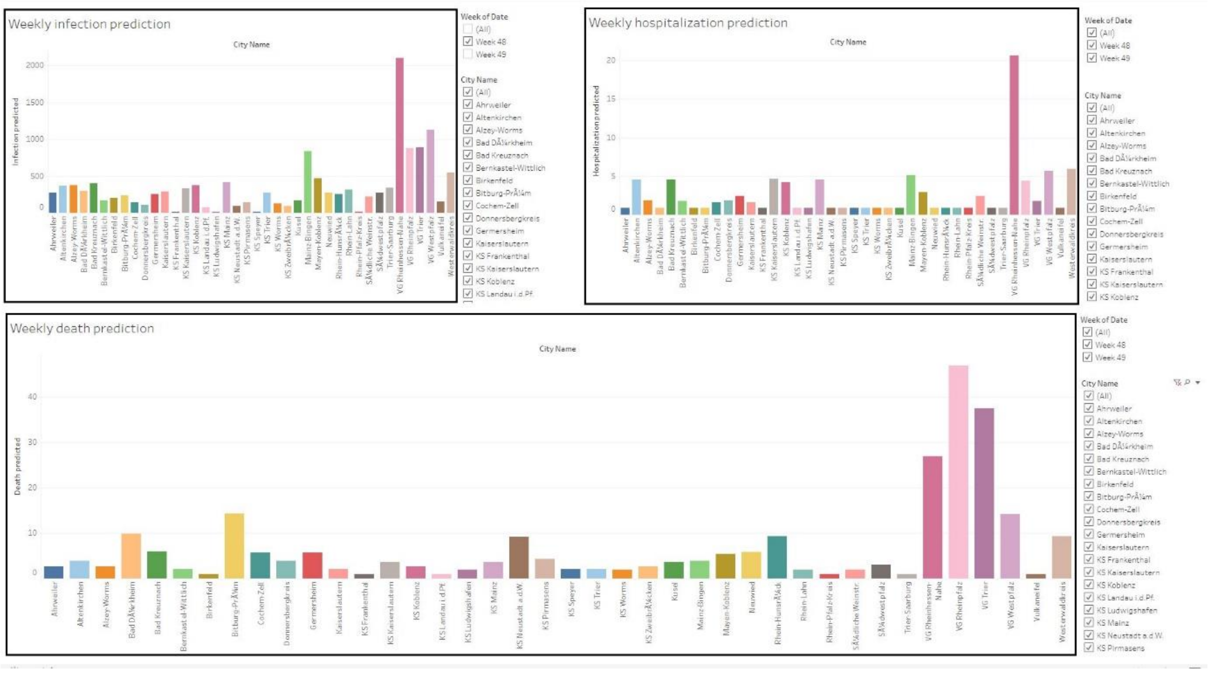


Figure 20 Dashboard for weekly predicted data

There are three predictions in the form of bar graphs visible on the dashboard, which are infection data, hospitalization data, and death data. There are filters on the cities and weeks. Users can easily select the desired city and week for a more narrowed-down view.

Web app creation and hosting on the internet

Author: Pathey Atulkumar Pandya

We made a web app which is showing the results of our findings visually and is available on the internet for everyone. There are several benefits to building web-based applications, including:

- Accessibility: Web applications can be accessed from any device with an internet connection, making them easily accessible to a wide range of users.
- Scalability: Web applications can be easily scaled to accommodate a large number of users, making them suitable for enterprise-level applications.
- Platform independence: Web applications can run on multiple platforms and devices, including desktops, laptops, tablets, and smartphones, without the need for separate versions or installations.
- Easy deployment: Web applications can be deployed and hosted on a variety of platforms and infrastructure, making it easy to distribute and share your application.
- Interactivity: Web applications can be built with interactive elements such as forms, buttons, and links which allow users to easily navigate and interact with the application.
- Real-time features: Web apps can be built to have real-time features such as notifications, messaging, and live updates, providing a more engaging and interactive user experience.

Here, we have decided to use the Streamlit library for the creation of the web app. Streamlit is a Python library that allows developers to create interactive web-based applications for data visualization and machine learning. Some of the main uses of Streamlit include:

- Building interactive dashboards for data exploration and analysis

- Creating user interfaces for machine learning models
- Building rapid prototyping tools for data scientists
- Building simple web-based tools for data visualization and exploration

Streamlit allows developers to create web-based applications quickly and easily with minimal coding and setup, making it a popular choice for data scientists and other developers working with data.

We were deciding to have a multi-pages web app. One page is showing the historic data and another one is for the prediction data.

For hosting our web app, we were choosing Streamlit cloud. It is a cloud-based platform, that allows you to deploy and share your Streamlit apps with others. First, we had to upload our code files on the Github repository. Then we had to connect our app to a GitHub repository which is dynamically updating the app whenever we are changing the code on the Github repository.

Once the app is deployed, we can share the URL with others to access the app.

Link: https://aicoviduniko.streamlit.app/

Integration and Testing

Author: Rahul Bhanushali and Pathey Atulkumar Pandya

Module Integration

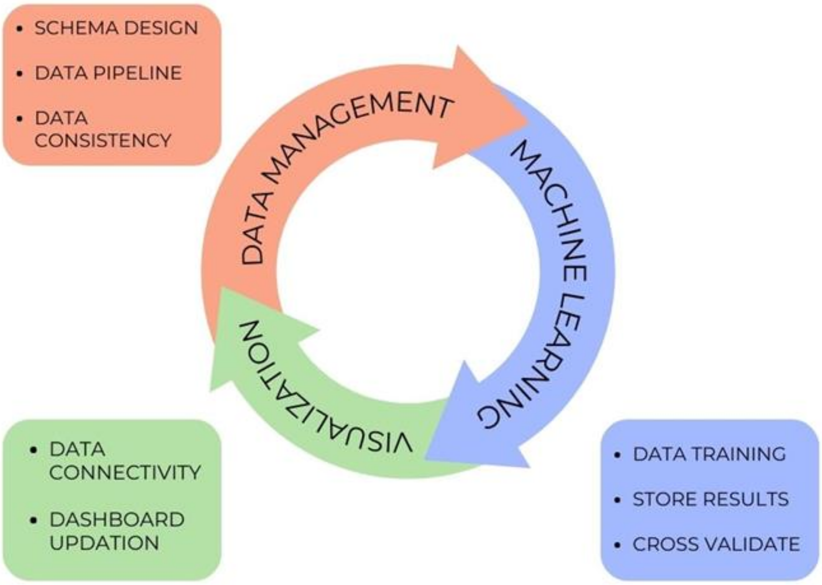


Figure 21 Module Integration

Module integration in a machine learning project can help to organize and structure code, making it easier to develop and maintain. By separating the code into different modules, we can focus on one aspect of the project at a time and avoid cluttering our main script with too much code. It is important to establish interaction between modules of Data Management, Machine Learning scripts, and Visualization.

To integrate the Tableau dashboard inside the Streamlit web app we were using an Html component. First, we had to publish our Tableau dashboard to Tableau Server or Tableau Online. Once it is published, we were able to use the embed code provided by Tableau to create an Html component in our Streamlit app. We used the components.html () function in Streamlit to embed the Tableau dashboard in our app. The function takes the URL of the Tableau dashboard and the width and height of the Html component as arguments. For example:

components.html (html_temp, height=1500, width=2000)

This method will only work if the Tableau dashboard is publicly accessible.

We need to ensure that we have installed any necessary dependencies and that the modules are compatible with each other and with the rest of our codebase. Module integration can provide below advantages in the project:

1. Code organization: By dividing the code into different modules, we can better organize the code and avoid cluttering the main script with too much code. This can make it easier to develop, debug, and maintain projects.
2. Reusability: It is easier to reuse the code in other projects. For example, we can have a data management module that contains functions for loading and pre-processing data, which can be used in multiple projects.

3. Maintainability: It is easier to update and maintain the project over time. If we need to make a change to one module, it is less likely to affect the rest of our codebase.
4. Collaboration: When working on a team, module integration can help to ensure that different team members can work on distinct parts of the project without stepping on each other's toes. Each team member can work on their own module, and then the modules can be combined and integrated into the final project.

Conclusion

Author: Md Naiem Siddique and Pathey Atulkumar Pandya

Collecting time variant historical COVID data was vital for our research. A complex interconnected dataset is necessary to build an expert system. In our research we have successfully implemented an automated data collection and processing engine. However, the data we collected and analysed lacks dimensionality. Several pharmaceuticals and non-pharmaceuticals intervention measures dictates the development of the virus infection. Some of the key intervention measures are vaccination, lockdown and social distancing, mask usage etc. Nevertheless, demographics, socio economic structure, characteristics and evolution of virus strains also contributes to the infection numbers.

Our machine learning models were rather simplistic. In order to accurately forecast infection waves we need to build complex models. We found ARIMA and SIRD model works better for future trend analysis compared to some other regression models. However, we need to incorporate age structure, vaccination, and seasonality into our model to improve accuracy. Moreover, a neural network can be helpful determining the model parameter from the existing large and complex dataset.

Future Scope of Work

Author: Rahul Bhanushali, Martin Okolie, and Pathey Atulkumar Pandya

1. One potential future scope for this project could be to acquire more COVID-19 data such as patient details, hospitalization data, and vaccination data. This would allow for a more comprehensive analysis of the pandemic, as these additional factors could provide valuable insights into the spread and impact of the virus.
2. To improve the performance and efficiency of the machine learning models, it may be beneficial to incorporate cross-validation techniques. This would allow for the models to be more robust and accurate, as they would be tested on a wider range of data.
3. Another potential avenue for future development would be to create user-friendly interactive dashboards with custom filters. This would allow users to access and analyse the data according to their specific requirements more easily. Overall, these efforts could help to further enhance our understanding of the COVID-19 pandemic and inform effective strategies for controlling its spread.
4. The models can also be improved by adding covariate values (values that are closely related to the target value) in the dataset during the machine learning forecasts. Values like this can be vaccination rates within a certain period of time.
5. Studying the effectiveness of non-pharmaceutical interventions (NPIs) by evaluating the effectiveness of different NPIs such as social distancing, mask-wearing, and quarantine measures in controlling the spread of the virus.
6. Investigating the long-term effects of the virus by studying the long-term health consequences of COVID-19, such as chronic fatigue, lung damage, and mental health issues.
7. Understanding the impact on health systems by examining the effects of the pandemic on healthcare systems, including the ability to meet the needs of patients, and identifying ways to improve preparedness for future outbreaks.

References

Breiman, L. (2001). *Random Forests* (Vol. 45).

Cao, Q., & Heydari, B. (2022). Micro-level social structures and the success of COVID-19 national policies. *Nature Computational Science*, *2*(9), 595–604. https://doi.org/10.1038/s43588-022-00314-0

Catboost pdf. (n.d.).

Davies, N. G., Klepac, P., Liu, Y., Prem, K., Jit, M., Pearson, C. A. B., Quilty, B. J., Kucharski, A. J., Gibbs, H., Clifford, S., Gimma, A., van Zandvoort, K., Munday, J. D., Diamond, C., Edmunds, W. J., Houben, R. M. G. J., Hellewell, J., Russell, T. W., Abbott, S., … Eggo, R. M. (2020). Age-dependent effects in the transmission and control of COVID-19 epidemics. *Nature Medicine*, *26*(8), 1205–1211. https://doi.org/10.1038/s41591-020-0962-9

Islam, M. N., Inan, T. T., Rafi, S., Akter, S. S., Sarker, I. H., & Islam, A. K. M. N. (2021). A Systematic Review on the Use of AI and ML for Fighting the COVID-19 Pandemic. *IEEE Transactions on Artificial Intelligence*, *1*(3), 258–270. https://doi.org/10.1109/tai.2021.3062771

Kermack, W. 0, & Mckendrick, A. G. (n.d.). *A Contribution to the Mathematical Theory of Epidemics*. https://royalsocietypublishing.org/

Marinov, T. T., & Marinova, R. S. (2022). Adaptive SIR model with vaccination: simultaneous identification of rates and functions illustrated with COVID-19. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-20276-7

Menda, K., Laird, L., Kochenderfer, M. J., & Caceres, R. S. (2021). Explaining COVID-19 outbreaks with reactive SEIRD models. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-97260-0

Oladunni, T., Denis, M., Ososanya, E., & Barry, A. (2021). Exponential smoothing forecast of African Americans' COVID-19 fatalities. *Proceedings - 2021 2nd International Conference on Computing and Data Science, CDS 2021*, 466–471. https://doi.org/10.1109/CDS52072.2021.00086

Saha, S., Samanta, G., & Nieto, J. J. (2022). Impact of optimal vaccination and social distancing on COVID-19 pandemic. *Mathematics and Computers in Simulation*, *200*, 285–314. https://doi.org/10.1016/j.matcom.2022.04.025

Spannaus, A., Papamarkou, T., Erwin, S., & Christian, J. B. (2022a). Inferring the spread of COVID-19: the role of time-varying reporting rate in epidemiological modelling. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-14979-0

Spannaus, A., Papamarkou, T., Erwin, S., & Christian, J. B. (2022b). Inferring the spread of COVID-19: the role of time-varying reporting rate in epidemiological modelling. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-14979-0

Srivastava, V., Srivastava, S., Chaudhary, G., & Al-Turjman, F. (2020). A systematic approach for COVID-19 predictions and parameter estimation. *Personal and Ubiquitous Computing*. https://doi.org/10.1007/s00779-020-01462-8

Sy, K. T. L., White, L. F., & Nichols, B. E. (2021). Population density and basic reproductive number of COVID-19 across United States counties. *PLoS ONE*, *16*(4 April). https://doi.org/10.1371/journal.pone.0249271

Syrowatka, A., Kuznetsova, M., Alsubai, A., Beckman, A. L., Bain, P. A., Craig, K. J. T., Hu, J., Jackson, G. P., Rhee, K., & Bates, D. W. (2021). Leveraging artificial intelligence for pandemic preparedness and response: a scoping review to identify key use cases. In *npj Digital Medicine* (Vol. 4, Issue 1). Nature Research. https://doi.org/10.1038/s41746-021-00459-8

Wangping, J., Ke, H., Yang, S., Wenzhe, C., Shengshu, W., Shanshan, Y., Jianwei, W., Fuyin, K., Penggang, T., Jing, L., Miao, L., & Yao, H. (2020). Extended SIR Prediction of the Epidemics Trend of COVID-19 in Italy and Compared with Hunan, China. *Frontiers in Medicine*, *7*. https://doi.org/10.3389/fmed.2020.00169

Wu, J. T., Leung, K., & Leung, G. M. (2020). Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. *The Lancet*, *395*(10225), 689–697. https://doi.org/10.1016/S0140-6736(20)30260-9

Zhao, D., Zhang, R., Zhang, H., & He, S. (2022). Prediction of global omicron pandemic using ARIMA, MLR, and Prophet models. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-23154-4