# Prediction of COVID-19 in North America

Student: Angela Amador

TMU Student Number: 500259095

Supervisor: Tamer Abdou, PhD

Submission Date: Nov 27th, 2023


Ryerson University

# Table of Contents

# Abstract

The Coronavirus disease (COVID-10) was first reported in December 2019 in Wuhan, Hubei Province, China. It created a calamitous situation throughout the world as cumulative incidents of COVID-19 rapidly increased day by day. In the absence of any medications, the only solution was to slow down the spread by exercising “social distancing” (hard lock-downs, restrictions on people mobility, limitations of the number of people in public places and the usage of protection gear (masks or gloves), among others) to block the chain of the spread of the virus. Here it is where Machine Learning models helped forecast where and when the disease was likely to spread, and support those regions, governance and entities on their decision making.

# Literature Review

Several publications and studies were reviewed with emphasis placed on predicting the number of cases around the world and how these Machine Learning (ML) models helped governments and other organizations to better prepare for the pandemic.

## Machine Learning-Based Research for COVID-19 Detection, Diagnosis, and Prediction: A Survey (Meraihi et al., 2022)

This paper reviews more than 160 Machine Learning based approaches developed to help with the pandemic. It addresses detection, diagnosis, and prediction approaches. From the scope of my project and based on the analysis of the paper, these are the methods and data types that have been used for prediction:

Some of the supervised learning models for prediction of COVID-19 cases:

| **Method Name** | **Data Type** |
|---|---|
| Support Vector Machine (SVM) with Decision Tree (DT) | X-ray image |
| Support Vector Machine (SVM) | Text |
| Least Square-SVM (LS-SVM) and Autoregressive Integrated Moving Average (ARIMA) | Time series |
| Linear regression model and Random Forest | CT images |
| Logistic regression model | CT images |
| XGBoost | Time series |
| Linear regression model with Support Vector Machine (SVM) Model and Artificial Neural Network (ANN) | Text |
| Linear regression and SEIR (Susceptible, Exposed, Infectious, Recovered) | Time series |
| Logistic Regression with Random Forest, Partial Least Squares Regression (PLSR), Elastic Net and Bagged Flexible Discriminant Analysis (BFDA) | Time series |
| Support Vector Regression (SVR), Stacking Ensemble Learning (SEL), | Time series |

| **Method Name** | **Data Type** |
|---|---|
| Auto-Regression Integrated Moving Average (ARIMA), Cubist Regression (CUBIST), Random Forest (RF), Ridge Regression (RIDGE) | |
| Support Vector Regression (SVR), Linear Regression and Polynomial Regression | Text |
| Linear regression models (Penalized Binomial Regression (PBR), Conditional Inference Trees (CIR), Generalised Linear (GL), and SVM with linear kernel) | CT Images and clinical data |
| PBRR (combination of Bayesian Ridge Regression (BRR) with n-degree Polynomial for forecasting) | Text |
| Fine-tuned Random Forest model with AdaBoost algorithm | Text |

Some of the Convolutional Neural Networks (CNN) approaches for prediction of COVID-19 cases:

| **Method Name** | **Data Type** |
|---|---|
| DenseNet-121 | CT images |

Some of the Recurrent Neural Networks (RNN) approaches for prediction of COVID-19 cases:

| **Method Name** | **Data Type** |
|---|---|
| LSTM with NLP | Text |
| LSTM | Text |
| LSTM | Time series |

Specialized CNN approaches for prediction:

| **Method Name** | **Data Type** |
|---|---|
| COVID–SDNet | X-ray images |

Other Machine Learning approaches for prediction of COVID-19 cases:

| **Method Name** | **Data Type** |
|---|---|
| Autoregressive Integrated Moving Average (ARIMA) model and Wavelet-based forecasting (WBF) model | Time series |
| MAchine learning and Cloud Computing | Time series |
| FbProphet technique and Logistic Model | Time series |
| Kalman Filter model | Text |

## Kalman filter based short term prediction model for COVID-19 spread (Singh et al., 2021)

This article analyzes various studies using data on the COVID-19 spread which includes demographic and environmental factors to be used into different ML Models like minimum temperature, maximum temperature, humidity, and rainfall in India.

Here, Kalman filter is used to forecast COVID19 incidence, and . Pearson correlation is used to find the dependencies among different features of the data. The importance of individual features in the proposed model is calculated through the random forest algorithm.

The article concludes the proposed prediction model is good for short term prediction i.e. daily and weekly. The proposed prediction model can be updated to further accommodate long term and medium term series prediction in future.

## Predicting the Growth and Trend of COVID-19 Pandemic using Machine Learning and Cloud Computing (Tuli et al., 2020)

The focus of this article in addition to ML is Cloud Computing and how its power helped with the process to develop, manage and analyse big data.  Cloud computing can be used to rapidly enhance the prediction process using high-speed computations.

The focus is to show that using iterative weighting for fitting Generalized Inverse Weibull (GIW) distribution, a better fit can be obtained to develop a prediction framework.

## Machine learning-based prediction of COVID-19 diagnosis based on symptoms (Zoabi et al., 2021)

This paper proposed a machine-learning model that predicts a positive SARS-CoV-2 infection in a RT-PCR test by asking eight basic questions. The model was trained on data of all individuals in Israel tested for SARS-CoV-2 during the first months of the COVID-19 pandemic. The model was implemented globally for effective screening and prioritization of testing for the virus in the general population.

Because the data is coming from surveys, it has limitations, biases and missing information. Training and testing a model while filtering out symptoms of high bias in advance still achieved very high accuracy. The methodology presented in this study may benefit the health system response to future epidemic waves of this disease and of other respiratory viruses in general.

Predictions were generated using a gradient-boosting machine model built with decision-tree base-learners.

## Forecast and prediction of COVID-19 using machine learning (Painuli et al., 2021)

The article discusses Auto Regressive Integrated Moving Average (ARIMA) time series for forecasting confirmed cases for various states in India. Two classifiers, Random Forest and Extra Tree Classifier (ETC), were selected. These results can be used to take corrective measures by different government bodies and assist with forecasting and planning in the fight against infectious diseases such as COVID-19.

## Forecasting COVID-19 spreading through an ensemble of classical and machine learning models: Spain’s case study (Heredia Cacha et al., 2023)

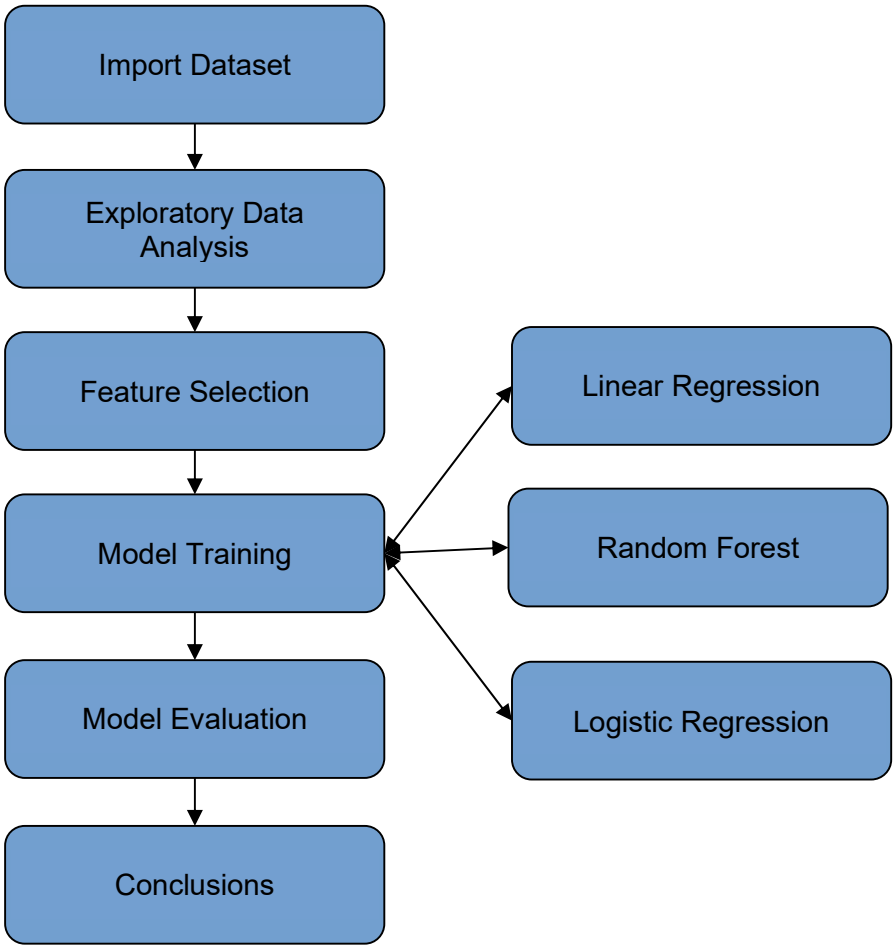
This article combines both ML and classical population models, using exclusively publicly available data of incidence, mobility, vaccination and weather in Spain.

In this work the performance of four ML models were evaluated (Random Forest, Gradient Boosting, k-Nearest Neighbors and Kernel Ridge Regression), and four population models (Gompertz, Logistic, Richards and Bertalanffy) in order to estimate the near future evolution of the COVID-19 pandemic, using daily cases data, together with vaccination, mobility and weather data.

## Conclusion

The COVID-19 pandemic affected everyone around the world and brought together researchers and investigation communities from different fields to apply multiple approaches and quickly learn about it. As result of this effort multiple approaches were applied using ML techniques to identify spread patterns, vulnerable demographics, effective social restrictions, and in general life-saving strategies. In this work I take the opportunity to duplicate the same strategies and replicate some of the research using the same data.

# Approach




# Exploratory Data Analysis

## Dataset

The dataset, provided by **Our World in Data**, provides COVID-19 vaccination information collected by **Our World in Data** and made available to the **Kaggle community** https://www.kaggle.com/datasets/caesarmario/our-world-in-data-covid19-dataset/download?datasetVersionNumber=418. This dataset is updated daily, and for the purpose of this study I am analyzing the data with information up to Oct 7th, 2023.

The dataset is a comma separated values file, with 67 variables and 346,567 observations.

## Data Dictionary

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 346567 entries, 0 to 346566
Data columns (total 67 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   iso_code                              346567 non-null  object
 1   continent                             330089 non-null  object
 2   location                              346567 non-null  object
 3   date                                  346567 non-null  object
 4   total_cases                           308672 non-null  float64
 5   new_cases                             337028 non-null  float64
 6   new_cases_smoothed                    335769 non-null  float64
 7   total_deaths                          287169 non-null  float64
 8   new_deaths                            337072 non-null  float64
 9   new_deaths_smoothed                   335842 non-null  float64
 10  total_cases_per_million               308672 non-null  float64
 11  new_cases_per_million                 337028 non-null  float64
 12  new_cases_smoothed_per_million        335769 non-null  float64
 13  total_deaths_per_million              287169 non-null  float64
 14  new_deaths_per_million                337072 non-null  float64
 15  new_deaths_smoothed_per_million       335842 non-null  float64
 16  reproduction_rate                     184817 non-null  float64
 17  icu_patients                          37509 non-null   float64
 18  icu_patients_per_million              37509 non-null   float64
 19  hosp_patients                         38759 non-null   float64
 20  hosp_patients_per_million             38759 non-null   float64
 21  weekly_icu_admissions                 10160 non-null   float64
 22  weekly_icu_admissions_per_million     10160 non-null   float64
 23  weekly_hosp_admissions                23145 non-null   float64
 24  weekly_hosp_admissions_per_million    23145 non-null   float64
 25  total_tests                           79387 non-null   float64
 26  new_tests                             75403 non-null   float64
 27  total_tests_per_thousand              79387 non-null   float64
 28  new_tests_per_thousand                75403 non-null   float64
 29  new_tests_smoothed                    103965 non-null  float64
 30  new_tests_smoothed_per_thousand       103965 non-null  float64
 31  positive_rate                         95927 non-null   float64
 32  tests_per_case                        94348 non-null   float64
 33  tests_units                           106788 non-null  object
 34  total_vaccinations                    78953 non-null   float64
 35  people_vaccinated                     75575 non-null   float64
```

```
 36  people_fully_vaccinated                          72224 non-null   float64
 37  total_boosters                                   47234 non-null   float64
 38  new_vaccinations                                 65019 non-null   float64
 39  new_vaccinations_smoothed                        180079 non-null  float64
 40  total_vaccinations_per_hundred                   78953 non-null   float64
 41  people_vaccinated_per_hundred                    75575 non-null   float64
 42  people_fully_vaccinated_per_hundred              72224 non-null   float64
 43  total_boosters_per_hundred                       47234 non-null   float64
 44  new_vaccinations_smoothed_per_million            180079 non-null  float64
 45  new_people_vaccinated_smoothed                   179887 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred       179887 non-null  float64
 47  stringency_index                                 197651 non-null  float64
 48  population_density                               294167 non-null  float64
 49  median_age                                       273580 non-null  float64
 50  aged_65_older                                    264005 non-null  float64
 51  aged_70_older                                    270838 non-null  float64
 52  gdp_per_capita                                   268118 non-null  float64
 53  extreme_poverty                                  172778 non-null  float64
 54  cardiovasc_death_rate                            268731 non-null  float64
 55  diabetes_prevalence                              282404 non-null  float64
 56  female_smokers                                   201575 non-null  float64
 57  male_smokers                                     198833 non-null  float64
 58  handwashing_facilities                           131627 non-null  float64
 59  hospital_beds_per_thousand                       237221 non-null  float64
 60  life_expectancy                                  318823 non-null  float64
 61  human_development_index                          260466 non-null  float64
 62  population                                       346567 non-null  float64
 63  excess_mortality_cumulative_absolute             11953 non-null   float64
 64  excess_mortality_cumulative                      11953 non-null   float64
 65  excess_mortality                                 11953 non-null   float64
 66  excess_mortality_cumulative_per_million          11953 non-null   float64
dtypes: float64(62), object(5)
memory usage: 177.2+ MB
```

## Metadata

The dataset size is 91.1 MB, the Pandas data profiling is almost 300 MB. This initial analysis can be found in GitHub: https://github.com/aamadorc/CIND820/blob/main/CIND820_EDA_DataProfiling.html. Due to the size of the resulting file, it is directly stored in Git LFS so it cannot be previewed, but instead it needs to be downloaded. However, a GitHub Action is set up in the repository to make every version of the resulting `ipynb` and `html` files browseable through the corresponding GitHub Pages site. The direct link to the data profiling of the complete dataset is https://aamadorc.github.io/CIND820/47631c816631ff3a8b42bb60ff824760cc50d6c9-CIND820_EDA_DataProfiling.html

Below is the metadata of the whole dataset.

| Variable | Description |
|---|---|
| iso_code | ISO 3166-1 alpha-3 – three-letter country codes. Note that OWID-defined regions (e.g. continents like 'Europe') contain prefix 'OWID_'. |
| continent | Continent of the geographical location |
| location | Geographical location |
| date | Date of observation |
| total_cases | Total confirmed cases of COVID-19. Counts can include probable cases, where reported. |
| new_cases | New confirmed cases of COVID-19. Counts can include probable cases, where reported. In rare cases where our source reports a negative daily change due to a data correction, we set this metric to NA. |
| new_cases_smoothed | New confirmed cases of COVID-19 (7-day smoothed). Counts can include probable cases, where reported. |
| total_deaths | Total deaths attributed to COVID-19. Counts can include probable deaths, where reported. |
| new_deaths | New deaths attributed to COVID-19. Counts can include probable deaths, where reported. In rare cases where our source reports a negative daily change due to a data correction, we set this metric to NA. |
| new_deaths_smoothed | New deaths attributed to COVID-19 (7-day smoothed). Counts can include probable deaths, where reported. |
| total_cases_per_million | Total confirmed cases of COVID-19 per 1,000,000 people. Counts can include probable cases, where reported. |
| new_cases_per_million | New confirmed cases of COVID-19 per 1,000,000 people. Counts can include probable cases, where reported. |
| new_cases_smoothed_per_million | New confirmed cases of COVID-19 (7-day smoothed) per 1,000,000 people. Counts can include probable cases, where reported. |
| total_deaths_per_million | Total deaths attributed to COVID-19 per 1,000,000 people. Counts can include probable deaths, where reported. |
| new_deaths_per_million | New deaths attributed to COVID-19 per 1,000,000 people. Counts can include probable deaths, where reported. |
| new_deaths_smooth | New deaths attributed to COVID-19 (7-day smoothed) per 1,000,000 |

| Variable | Description |
|---|---|
| ed_per_million | people. Counts can include probable deaths, where reported. |
| reproduction_rate | Real-time estimate of the effective reproduction rate (R) of COVID-19. See https://github.com/crondonm/TrackingR/tree/main/Estimates-Database |
| icu_patients | Number of COVID-19 patients in intensive care units (ICUs) on a given day |
| icu_patients_per_million | Number of COVID-19 patients in intensive care units (ICUs) on a given day per 1,000,000 people |
| hosp_patients | Number of COVID-19 patients in hospital on a given day |
| hosp_patients_per_million | Number of COVID-19 patients in hospital on a given day per 1,000,000 people |
| weekly_icu_admissions | Number of COVID-19 patients newly admitted to intensive care units (ICUs) in a given week (reporting date and the preceeding 6 days) |
| weekly_icu_admissions_per_million | Number of COVID-19 patients newly admitted to intensive care units (ICUs) in a given week per 1,000,000 people (reporting date and the preceding 6 days) |
| weekly_hosp_admissions | Number of COVID-19 patients newly admitted to hospitals in a given week (reporting date and the preceding 6 days) |
| weekly_hosp_admissions_per_million | Number of COVID-19 patients newly admitted to hospitals in a given week per 1,000,000 people (reporting date and the preceding 6 days) |
| total_tests | Total tests for COVID-19 |
| new_tests | New tests for COVID-19 (only calculated for consecutive days) |
| total_tests_per_thousand | Total tests for COVID-19 per 1,000 people |
| new_tests_per_thousand | New tests for COVID-19 per 1,000 people |
| new_tests_smoothed | New tests for COVID-19 (7-day smoothed). For countries that don't report testing data on a daily basis, we assume that testing changed equally on a daily basis over any periods in which no data was reported. This produces a complete series of daily figures, which is then averaged over a rolling 7-day window |

| Variable | Description |
|---|---|
| new_tests_smoothed_per_thousand | New tests for COVID-19 (7-day smoothed) per 1,000 people |
| positive_rate | The share of COVID-19 tests that are positive, given as a rolling 7-day average (this is the inverse of tests_per_case) |
| tests_per_case | Tests conducted per new confirmed case of COVID-19, given as a rolling 7-day average (this is the inverse of positive_rate) |
| tests_units | Units used by the location to report its testing data. A country file can't contain mixed units. All metrics concerning testing data use the specified test unit. Valid units are 'people tested' (number of people tested), 'tests performed' (number of tests performed. a single person can be tested more than once in a given day) and 'samples tested' (number of samples tested. In some cases, more than one sample may be required to perform a given test.) |
| total_vaccinations | Total number of COVID-19 vaccination doses administered |
| people_vaccinated | Total number of people who received at least one vaccine dose |
| people_fully_vaccinated | Total number of people who received all doses prescribed by the initial vaccination protocol |
| total_boosters | Total number of COVID-19 vaccination booster doses administered (doses administered beyond the number prescribed by the vaccination protocol) |
| new_vaccinations | New COVID-19 vaccination doses administered (only calculated for consecutive days) |
| new_vaccinations_smoothed | New COVID-19 vaccination doses administered (7-day smoothed). For countries that don't report vaccination data on a daily basis, we assume that vaccination changed equally on a daily basis over any periods in which no data was reported. This produces a complete series of daily figures, which is then averaged over a rolling 7-day window |
| total_vaccinations_per_hundred | Total number of COVID-19 vaccination doses administered per 100 people in the total population |
| people_vaccinated_per_hundred | Total number of people who received at least one vaccine dose per 100 people in the total population |

| Variable | Description |
|---|---|
| people_fully_vaccinated_per_hundred | Total number of people who received all doses prescribed by the initial vaccination protocol per 100 people in the total population |
| total_boosters_per_hundred | Total number of COVID-19 vaccination booster doses administered per 100 people in the total population |
| new_vaccinations_smoothed_per_million | New COVID-19 vaccination doses administered (7-day smoothed) per 1,000,000 people in the total population |
| new_people_vaccinated_smoothed | Daily number of people receiving their first vaccine dose (7-day smoothed) |
| new_people_vaccinated_smoothed_per_hundred | Daily number of people receiving their first vaccine dose (7-day smoothed) per 100 people in the total population |
| stringency_index | Government Response Stringency Index: composite measure based on 9 response indicators including school closures, workplace closures, and travel bans, rescaled to a value from 0 to 100 (100 = strictest response) |
| population_density | Number of people divided by land area, measured in square kilometers, most recent year available |
| median_age | Median age of the population, UN projection for 2020 |
| aged_65_older | Share of the population that is 65 years and older, most recent year available |
| aged_70_older | Share of the population that is 70 years and older in 2015 |
| gdp_per_capita | Gross domestic product at purchasing power parity (constant 2011 international dollars), most recent year available |
| extreme_poverty | Share of the population living in extreme poverty, most recent year available since 2010 |
| cardiovasc_death_rate | Death rate from cardiovascular disease in 2017 (annual number of deaths per 100,000 people) |
| diabetes_prevalence | Diabetes prevalence (% of population aged 20 to 79) in 2017 |
| female_smokers | Share of women who smoke, most recent year available |

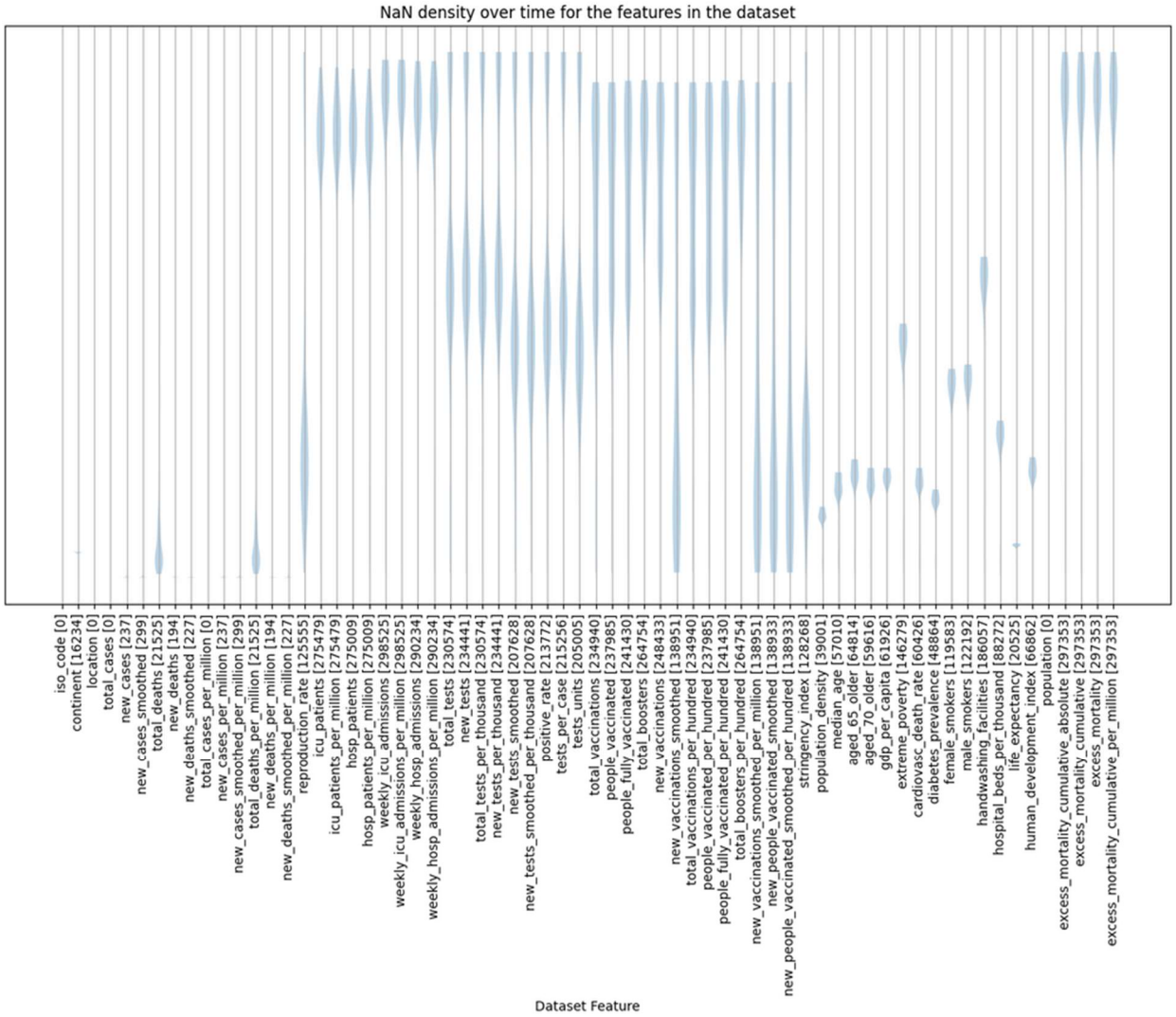
| Variable | Description |
|---|---|
| male_smokers | Share of men who smoke, most recent year available |
| handwashing_facilities | Share of the population with basic handwashing facilities on premises, most recent year available |
| hospital_beds_per_thousand | Hospital beds per 1,000 people, most recent year available since 2010 |
| life_expectancy | Life expectancy at birth in 2019 |
| human_development_index | A composite index measuring average achievement in three basic dimensions of human development—a long and healthy life, knowledge and a decent standard of living. Values for 2019, imported from http://hdr.undp.org/en/indicators/137506 |
| population | Population (latest available values). See https://github.com/owid/covid-19-data/blob/master/scripts/input/un/population_latest.csv for full list of sources |
| excess_mortality_cumulative_absolute | Cumulative difference between the reported number of deaths since 1 January 2020 and the projected number of deaths for the same period based on previous years. For more information, see https://github.com/owid/covid-19-data/tree/master/public/data/excess_mortality |
| excess_mortality_cumulative | Percentage difference between the cumulative number of deaths since 1 January 2020 and the cumulative projected deaths for the same period based on previous years. For more information, see https://github.com/owid/covid-19-data/tree/master/public/data/excess_mortality |
| excess_mortality | Percentage difference between the reported number of weekly or monthly deaths in 2020–2021 and the projected number of deaths for the same period based on previous years. For more information, see https://github.com/owid/covid-19-data/tree/master/public/data/excess_mortality |
| excess_mortality_cumulative_per_million | Cumulative difference between the reported number of deaths since 1 January 2020 and the projected number of deaths for the same period based on previous years, per million people. For more information, see https://github.com/owid/covid-19-data/tree/master/public/data/excess_mortality |

The overview of the initial dataset is:

| Dataset statistics | | Variable types | |
|---|---|---|---|
| **Number of variables** | 67 | **Text** | 2 |
| **Number of observations** | 346567 | **Categorical** | 2 |
| **Missing cells** | 11380440 | **DateTime** | 1 |
| **Missing cells (%)** | 49.0% | **Numeric** | 62 |
| **Duplicate rows** | 0 | | |
| **Duplicate rows (%)** | 0.0% | | |
| **Total size in memory** | 177.2 MiB | | |
| **Average record size in memory** | 536.0 B | | |

The dataset has a total of 37,895 observations with `NaN` values for the `total_cases` column. As this is my predicted feature, `NaN`s don't add value to the model so these observations were removed.

This is the `NaN` density over time for the features in the dataset:




## Limiting the scope of the model to North America

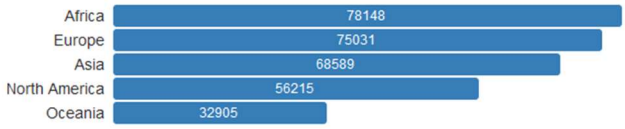
Due to the size of the file, and the large number of different values for the `iso_code` and `location` features, I had to make the decision of dropping data from outside North America. Once I apply One Hot Encoding, the dataset is left with 552 attributes.

The initial data profiling for `continent` is:

continent
Categorical

HIGH CORRELATION MISSING

| | |
|---|---|
| Distinct | 6 |
| Distinct (%) | < 0.1% |
| Missing | 16478 |
| Missing (%) | 4.8% |
| Memory size | 2.6 MiB |




After removing observations from outside of North America and without `NaN` in `total_cases`, we have a total of 53,290 observations.

### Remove data pre-dating COVID vaccine availability

Multiple vaccines became available in the second semester of 2020. By December most countries had access to approved vaccines.

As vaccinations changed the behaviour of the pandemic I removed data before January 1st, 2021 and considered only data collected after vaccines became available.

# Feature Selection

## One Hot Encoding in Machine Learning

Machine Learning models do not work with categorical data. To fit features with categorical data into the machine learning model, it needs to be converted first into numerical data. One technique for this is One Hot Encoding.

We have four categorical variables in the dataset: `iso_code`, `continent`, `location` and `tests_units`.

| Feature | Number of unique values |
|---|---|
| iso_code | 41 |
| continent | 1 |

| | |
|---|---|
| location | 41 |
| tests_units | 4 |

These are the statistics after running One Hot Encoding to the categorical variables:

| One hot encoding categorical variables: | Initially | iso_code | continent | location | test_units |
|---|---|---|---|---|---|
| Number of observations | 41,287 | 41,287 | 41,287 | 41,287 | 41,287 |
| Number of attributes | 67 | 108 | 109 | 150 | 153 |
| Size | 2,766,229 | 4,458,996 | 4,500,283 | 6,193,050 | 6,316,911 |

Now, I can eliminate the original categorical attributes.

**Convert attribute `date` to epoch**

The dataset also includes a feature `date` in calendar format which is not numerical, and on which I tried different strategies.

I tried to use it as an index and use `date` as criteria to split the dataset, but this affected the accuracy of the models, see results here https://aamadorc.github.io/CIND820/3f5e3ad354050fd83a5a352fe15caefbf26f129f-CIND820_EDA.html.

## Summary of changes on the dataset

| Action | # Observations | # Attributes | Size |
|---|---|---|---|
| Original dataset | 346,567 | 67 | 23,219,989 |
| Eliminate records with `NaN` in `total_cases` | 308,672 | 67 | 20,681,024 |
| Eliminate records from outside North America | 53,290 | 67 | 3,570,430 |

| Action | # Observations | # Attributes | Size |
|---|---|---|---|
| Eliminate records pre-vaccine | 41,287 | 67 | 2,766,229 |
| After One Hot Encoding | 41,287 | 149 | 6,151,763 |

## Data Splitting

One of the first decisions to make is how to utilize the existing data. One common technique is to split the data into two groups typically referred to as the Training and Testing sets. The Training set is used to develop models and feature sets; it is the substrate for estimating parameters, comparing models, and all of the other activities required to reach a final model. The Testing set is used only at the conclusion of these activities for estimating a final, unbiased assessment of the model's performance. It is critical that the Testing set is not used prior to this point. Looking at the Testing set results would bias the outcomes since the Testing data will have become part of the model development process. (Kuhn & Johnson, 2020)

After applying random selection using `train_test_split` with 70% random selection for training dataset and 30% random selection for testing dataset:

| | Source dataset | Training data (sub)set | Testing data (sub)set |
|---|---|---|---|
| Number of observations | 41,287 | 28,900 | 12,387 |
| Number of attributes | 149 | 149 | 149 |
| Size | 6,151,763 | 4,306,100 | 1,845,663 |

## Data Cleaning and Dimensionality Reduction

Data cleaning will take place only on the training dataset excluding predictive attribute `total_cases`.

There are seven techniques for Dimensionality Reduction: Missing Values, Low Variance Filter, High Correlation Filter, PCA, Random Forests, Backward Feature Elimination, and Forward Feature Construction. (Silipo et al., 2014)

## Identify Columns That Contain a Single Value

The feature `continent` contains only one value which is North America, then this attribute can be eliminated as it doesn't affect or influence the prediction.

## Remove data columns with too many NaN values

We can calculate the ratio of missing values using a simple formula. The formula is the number of missing values in each column divided by the total number of observations. Generally, we can drop variables having a missing-value ratio of more than 60% or 70%. For my purpose I am going to use a threshold of 60% missing values and remove those attributes.

Attributes with more than 60.0% of missing values:

| | column | nan_count | nan_rate |
|---|---|---|---|
| 18 | weekly_icu_admissions_per_million | 28900 | 1.000000 |
| 17 | weekly_icu_admissions | 28900 | 1.000000 |
| 19 | weekly_hosp_admissions | 28200 | 0.975779 |
| 20 | weekly_hosp_admissions_per_million | 28200 | 0.975779 |
| 61 | excess_mortality_cumulative_per_million | 28095 | 0.972145 |
| 60 | excess_mortality | 28095 | 0.972145 |
| 59 | excess_mortality_cumulative | 28095 | 0.972145 |
| 58 | excess_mortality_cumulative_absolute | 28095 | 0.972145 |
| 13 | icu_patients | 27515 | 0.952076 |
| 16 | hosp_patients_per_million | 27515 | 0.952076 |
| 15 | hosp_patients | 27515 | 0.952076 |

| | column | nan_count | nan_rate |
|---|---|---|---|
| 14 | icu_patients_per_million | 27515 | 0.952076 |
| 38 | total_boosters_per_hundred | 26212 | 0.906990 |
| 32 | total_boosters | 26212 | 0.906990 |
| 22 | new_tests | 24918 | 0.862215 |
| 24 | new_tests_per_thousand | 24918 | 0.862215 |
| 33 | new_vaccinations | 24843 | 0.859619 |
| 21 | total_tests | 24483 | 0.847163 |
| 23 | total_tests_per_thousand | 24483 | 0.847163 |
| 31 | people_fully_vaccinated | 23331 | 0.807301 |
| 37 | people_fully_vaccinated_per_hundred | 23331 | 0.807301 |
| 36 | people_vaccinated_per_hundred | 23145 | 0.800865 |
| 30 | people_vaccinated | 23145 | 0.800865 |
| 29 | total_vaccinations | 23048 | 0.797509 |
| 35 | total_vaccinations_per_hundred | 23048 | 0.797509 |
| 28 | tests_per_case | 23039 | 0.797197 |
| 27 | positive_rate | 23007 | 0.796090 |
| 25 | new_tests_smoothed | 22710 | 0.785813 |
| 26 | new_tests_smoothed_per_thousand | 22710 | 0.785813 |

| | column | nan_count | nan_rate |
|---|---|---|---|
| 48 | extreme_poverty | 21111 | 0.730484 |
| 52 | male_smokers | 20410 | 0.706228 |
| 51 | female_smokers | 19702 | 0.681730 |
| 53 | handwashing_facilities | 19670 | 0.680623 |
| 42 | stringency_index | 18185 | 0.629239 |

## Feature Selection - Correlation and P-value (Vishal, 2022)

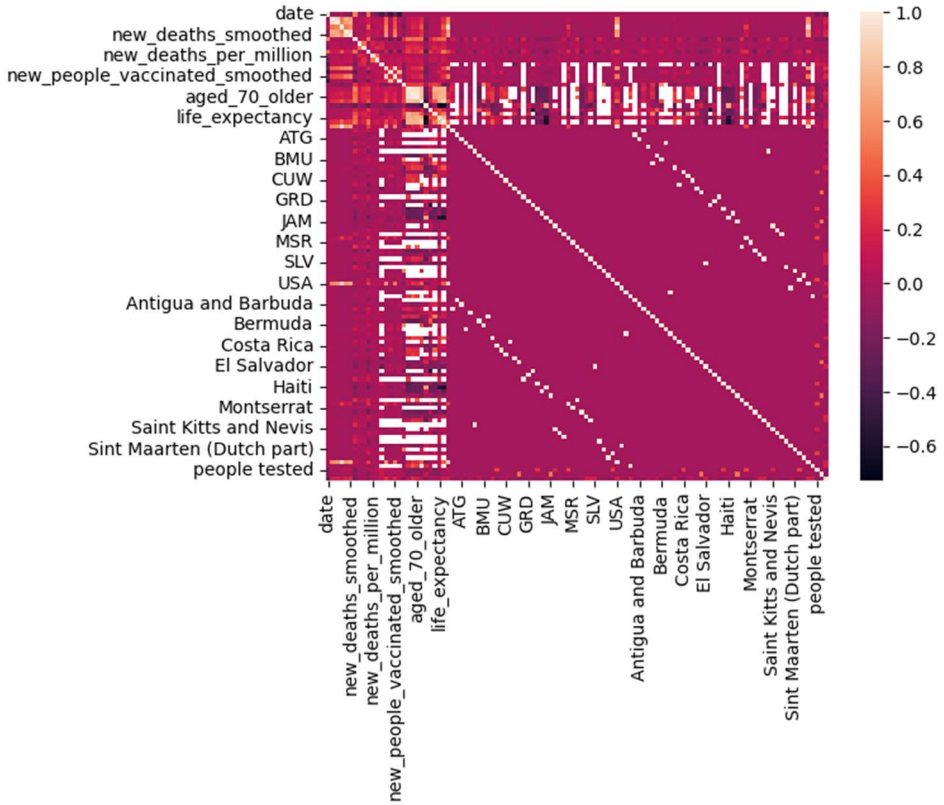
From (Vishal, 2022):

***How does correlation help in feature selection?***
*Features with high correlation are more linearly dependent and hence have almost the same effect on the dependent variable. So, when two features have high correlation, we can drop one of the two features.*

***What is p-value?***
*The P-value, probability value or asymptotic significance is a probability value for a given statistical model that, if the null hypothesis is true, a set of statistical observations more commonly known as the statistical summary is greater than or equal in magnitude to the observed results.*

***How does p-value help in feature selection?***
*Removal of different features from the dataset will have different effects on the p-value for the dataset. We can remove different features and measure the p-value in each case. These measured p-values can be used to decide whether to keep a feature or not.*

Using Feature Selection this is the resulting Correlation Heatmap:

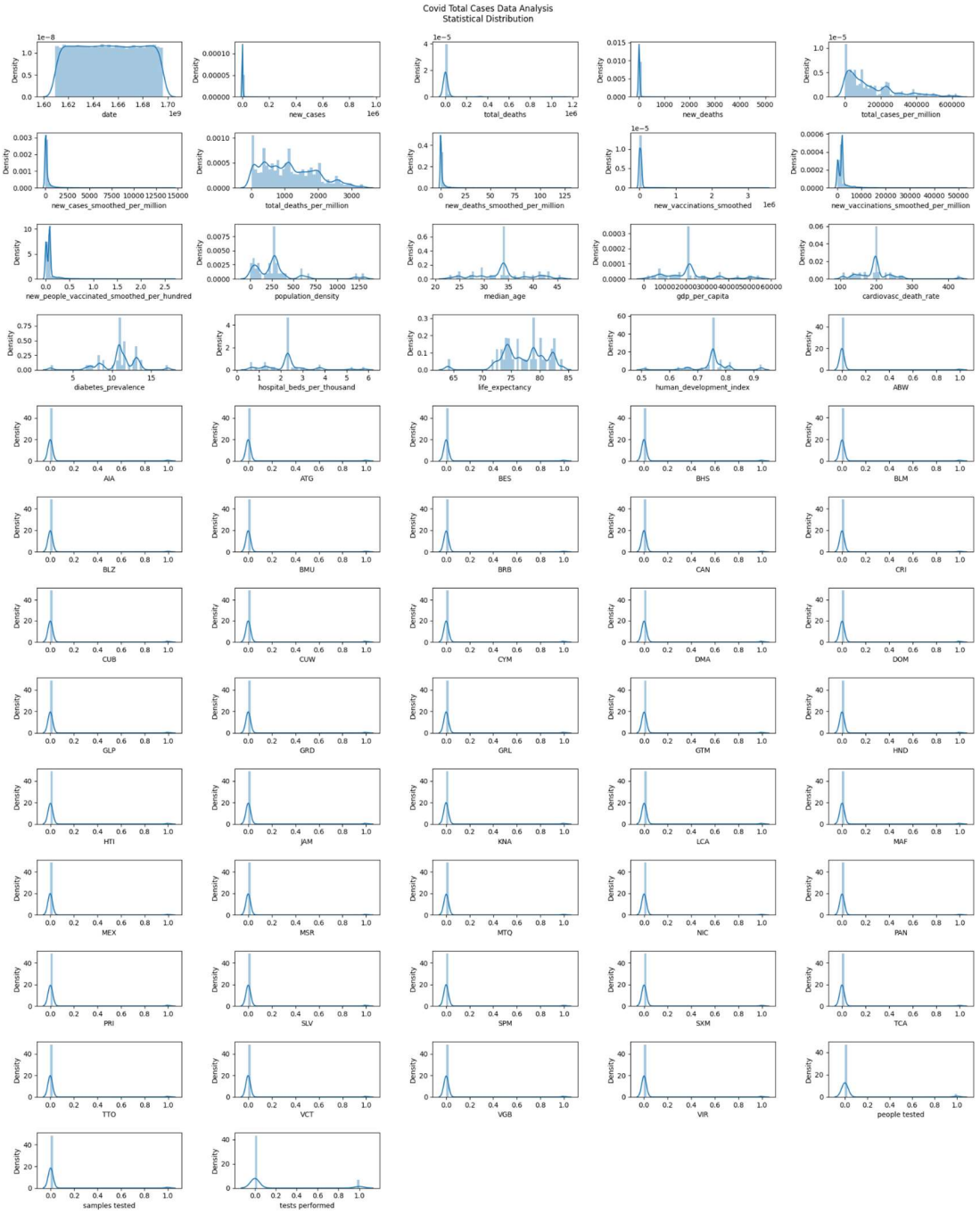



A total of 48 attributes with correlation higher than 0.9 were deleted.

Now, selecting columns based on p_value, for this we are going to use Backwards Elimination with a SL = 0.05. Two columns were selected to be eliminated.

## Visualize the selected features

Statistical distribution of the values for each one of the features:

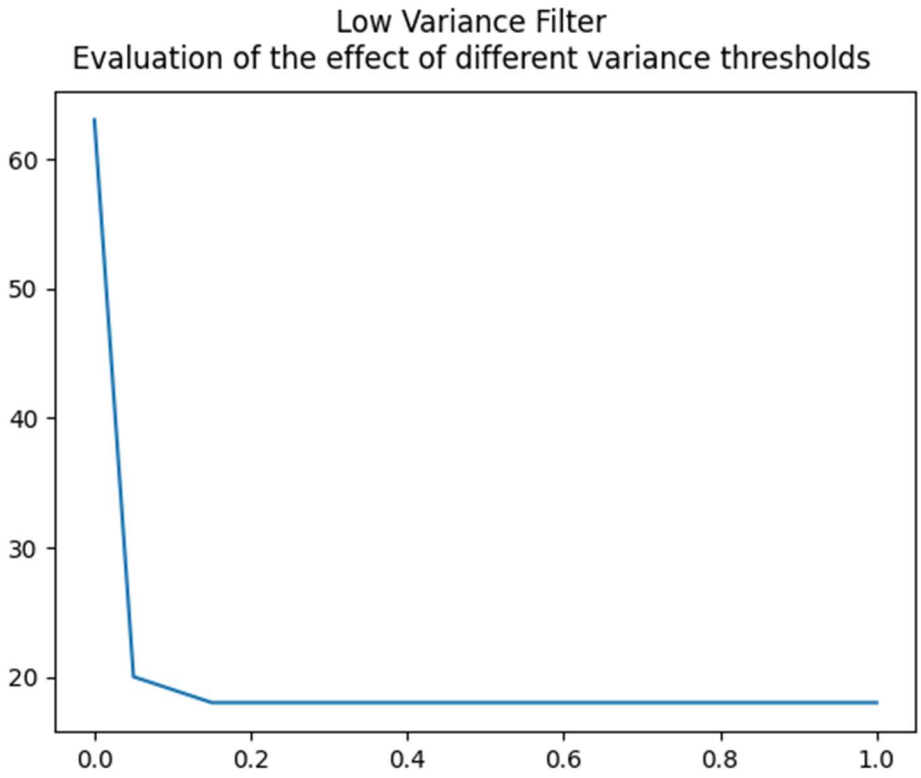


## Low Variance Filter

Another way of measuring how much information a feature has is to measure its variance. In the limit case where the feature assumes a constant value, the variance would be 0 and the column would be of no help in the discrimination of different groups of data.

The Low Variance Filter calculates each column variance and removes those columns with a variance value below a given threshold. However the variance can only be calculated for numerical columns, i.e. this dimensionality reduction method applies only to numerical columns as the variance value depends on the column numerical range. Therefore feature ranges need to be normalized to make variance values independent from the column domain range.

This plot illustrates the effect of different variance thresholds:




The line plot shows the relationship between the threshold and the number of features in the transformed dataset. I can see that with a small threshold of 0.15, 45 features are removed immediately.

### Summary of Dimensional Reduction

| Action | # Observations | # Attributes | Size |
|---|---|---|---|
| Original training dataset | 28,900 | 148 | 4,277,200 |
| Eliminate column(s) with single value | 28,900 | 147 | 4,248,300 |
| Eliminate columns with more than 60% `NaN` | 28,900 | 113 | 3,265,700 |
| Eliminate columns with correlation > 0.9 | 28,900 | 65 | 1,878,500 |
| After applying p_value and correlation | 28,900 | 63 | 1,820,700 |
| Eliminating columns with variance close to 0 | 28,900 | 18 | 520,200 |

## Data subset

Metadata of the subset, 18 variables:

- `date`
- `total_cases`
- `new_cases`
- `total_deaths`
- `new_deaths`
- `total_cases_per_million`
- `new_cases_smoothed_per_million`
- `total_deaths_per_million`
- `new_deaths_smoothed_per_million`
- `new_vaccinations_smoothed`
- `new_vaccinations_smoothed_per_million`
- `population_density`
- `median_age`
- `gdp_per_capita`
- `cardiovasc_death_rate`
- `diabetes_prevalence`
- `life_expectancy`
- `hospital_beds_per_thousand`

After analyzing the features that were eliminated and kept, data like testing doesn't add value in the prediction of the number of cases. As well as derived columns like weekly columns are not relevant. ICU information doesn't affect the number of cases either.

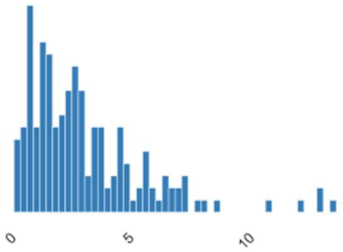
Also, it is interesting to see how information like gross domestic product at purchasing power, and cardiovascular and diabetes are pretty relevant to influence the number of cases.

Analysis on the selected attributes:

### hospital_beds_per_thousand
Real number (R)

HIGH CORRELATION MISSING

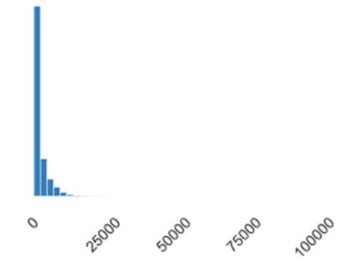
| | | | |
|---|---|---|---|
| Distinct | 102 | Minimum | 0.1 |
| Distinct (%) | 0.1% | Maximum | 13.8 |
| Missing | 80928 | Zeros | 0 |
| Missing (%) | 31.7% | Zeros (%) | 0.0% |
| Infinite | 0 | Negative | 0 |
| Infinite (%) | 0.0% | Negative (%) | 0.0% |
| Mean | 3.0971419 | Memory size | 12.0 MiB |




### new_vaccinations_smoothed_per_million
Real number (R)

MISSING ZEROS

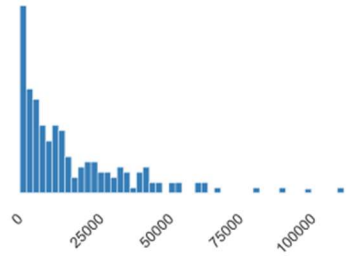
| | | | |
|---|---|---|---|
| Distinct | 12647 | Minimum | 0 |
| Distinct (%) | 7.0% | Maximum | 117113 |
| Missing | 75551 | Zeros | 3987 |
| Missing (%) | 29.6% | Zeros (%) | 1.6% |
| Infinite | 0 | Negative | 0 |
| Infinite (%) | 0.0% | Negative (%) | 0.0% |
| Mean | 1992.9024 | Memory size | 12.0 MiB |




### gdp_per_capita
Real number (R)

HIGH CORRELATION MISSING

| | | | |
|---|---|---|---|
| Distinct | 196 | Minimum | 661.24 |
| Distinct (%) | 0.1% | Maximum | 116935.6 |
| Missing | 58011 | Zeros | 0 |
| Missing (%) | 22.7% | Zeros (%) | 0.0% |
| Infinite | 0 | Negative | 0 |
| Infinite (%) | 0.0% | Negative (%) | 0.0% |
| Mean | 19076.009 | Memory size | 12.0 MiB |

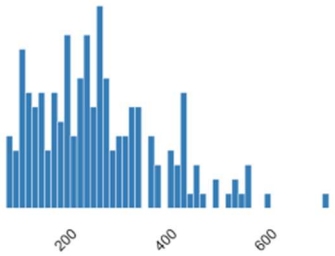


### cardiovasc_death_rate

Real number (R)

HIGH CORRELATION MISSING

| | | | |
|---|---|---|---|
| Distinct | 196 | Minimum | 79.37 |
| Distinct (%) | 0.1% | Maximum | 724.417 |
| Missing | 57777 | Zeros | 0 |
| Missing (%) | 22.6% | Zeros (%) | 0.0% |
| Infinite | 0 | Negative | 0 |
| Infinite (%) | 0.0% | Negative (%) | 0.0% |
| Mean | 264.27143 | Memory size | 12.0 MiB |

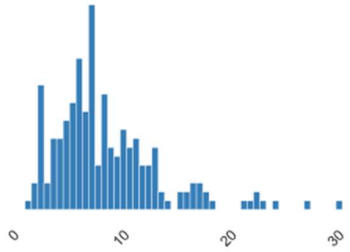



### diabetes_prevalence

Real number (R)

MISSING

| | | | |
|---|---|---|---|
| Distinct | 150 | Minimum | 0.99 |
| Distinct (%) | 0.1% | Maximum | 30.53 |
| Missing | 47729 | Zeros | 0 |
| Missing (%) | 18.7% | Zeros (%) | 0.0% |
| Infinite | 0 | Negative | 0 |
| Infinite (%) | 0.0% | Negative (%) | 0.0% |
| Mean | 8.5609601 | Memory size | 12.0 MiB |

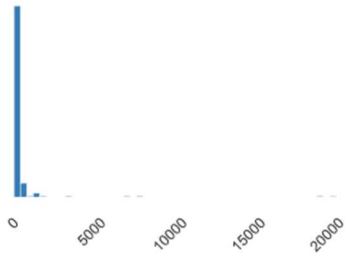



### population_density

Real number (R)

MISSING

| | | | |
|---|---|---|---|
| Distinct | 215 | Minimum | 0.137 |
| Distinct (%) | 0.1% | Maximum | 20546.766 |
| Missing | 38878 | Zeros | 0 |
| Missing (%) | 15.2% | Zeros (%) | 0.0% |
| Infinite | 0 | Negative | 0 |
| Infinite (%) | 0.0% | Negative (%) | 0.0% |
| Mean | 424.20264 | Memory size | 12.0 MiB |

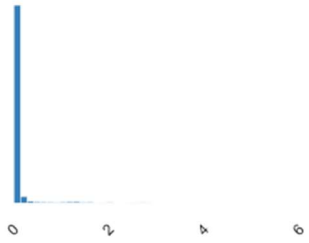


## total_deaths

Real number (R)

HIGH CORRELATION MISSING

| | | | |
|---|---|---|---|
| **Distinct** | 47081 | **Minimum** | 1 |
| **Distinct (%)** | 20.4% | **Maximum** | 6960770 |
| **Missing** | 24306 | **Zeros** | 0 |
| **Missing (%)** | 9.5% | **Zeros (%)** | 0.0% |
| **Infinite** | 0 | **Negative** | 0 |
| **Infinite (%)** | 0.0% | **Negative (%)** | 0.0% |
| **Mean** | 102124.29 | **Memory size** | 12.0 MiB |

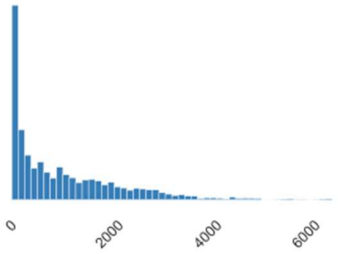



## total_deaths_per_million

Real number (R)

HIGH CORRELATION MISSING

| | | | |
|---|---|---|---|
| **Distinct** | 80704 | **Minimum** | 0.06 |
| **Distinct (%)** | 35.0% | **Maximum** | 6511.209 |
| **Missing** | 24306 | **Zeros** | 0 |
| **Missing (%)** | 9.5% | **Zeros (%)** | 0.0% |
| **Infinite** | 0 | **Negative** | 0 |
| **Infinite (%)** | 0.0% | **Negative (%)** | 0.0% |
| **Mean** | 1046.3662 | **Memory size** | 12.0 MiB |

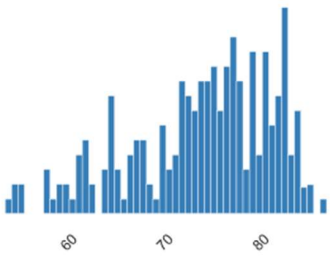



## life_expectancy

Real number (R)

HIGH CORRELATION MISSING

| | | | |
|---|---|---|---|
| **Distinct** | 220 | **Minimum** | 53.28 |
| **Distinct (%)** | 0.1% | **Maximum** | 86.75 |
| **Missing** | 20761 | **Zeros** | 0 |
| **Missing (%)** | 8.1% | **Zeros (%)** | 0.0% |
| **Infinite** | 0 | **Negative** | 0 |
| **Infinite (%)** | 0.0% | **Negative (%)** | 0.0% |
| **Mean** | 73.724505 | **Memory size** | 12.0 MiB |



### new_cases
Real number (R)

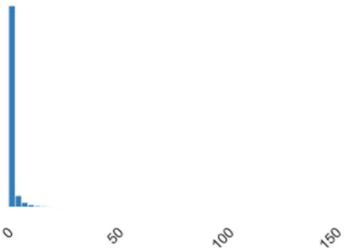
HIGH CORRELATION MISSING SKEWED ZEROS

| | | | |
|---|---|---|---|
| **Distinct** | 24233 | **Minimum** | 0 |
| **Distinct (%)** | 9.8% | **Maximum** | 8401961 |
| **Missing** | 7651 | **Zeros** | 117681 |
| **Missing (%)** | 3.0% | **Zeros (%)** | 46.1% |
| **Infinite** | 0 | **Negative** | 0 |
| **Infinite (%)** | 0.0% | **Negative (%)** | 0.0% |
| **Mean** | 11810.923 | **Memory size** | 12.0 MiB |


### new_deaths_smoothed_per_million
Real number (R)

HIGH CORRELATION MISSING ZEROS

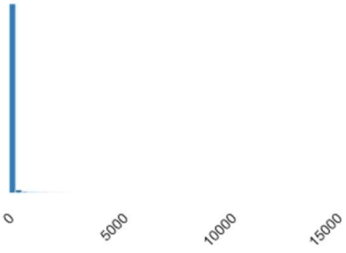
| | | | |
|---|---|---|---|
| **Distinct** | 9492 | **Minimum** | 0 |
| **Distinct (%)** | 3.8% | **Maximum** | 148.641 |
| **Missing** | 7636 | **Zeros** | 109651 |
| **Missing (%)** | 3.0% | **Zeros (%)** | 43.0% |
| **Infinite** | 0 | **Negative** | 0 |
| **Infinite (%)** | 0.0% | **Negative (%)** | 0.0% |
| **Mean** | 0.98019728 | **Memory size** | 12.0 MiB |




### new_deaths_smoothed
Real number (R)

HIGH CORRELATION MISSING ZEROS

| | | | |
|---|---|---|---|
| **Distinct** | 9454 | **Minimum** | 0 |
| **Distinct (%)** | 3.8% | **Maximum** | 14821.857 |
| **Missing** | 7636 | **Zeros** | 109279 |
| **Missing (%)** | 3.0% | **Zeros (%)** | 42.8% |
| **Infinite** | 0 | **Negative** | 0 |
| **Infinite (%)** | 0.0% | **Negative (%)** | 0.0% |
| **Mean** | 85.358373 | **Memory size** | 12.0 MiB |

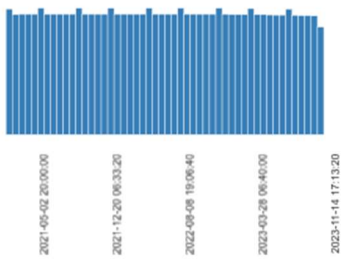


## date

Date

| | |
|---|---|
| Distinct | 1010 |
| Distinct (%) | 0.4% |
| Missing | 0 |
| Missing (%) | 0.0% |
| Memory size | 12.0 MiB |

| | |
|---|---|
| Minimum | 2021-01-01 00:00:00 |
| Maximum | 2023-10-07 00:00:00 |



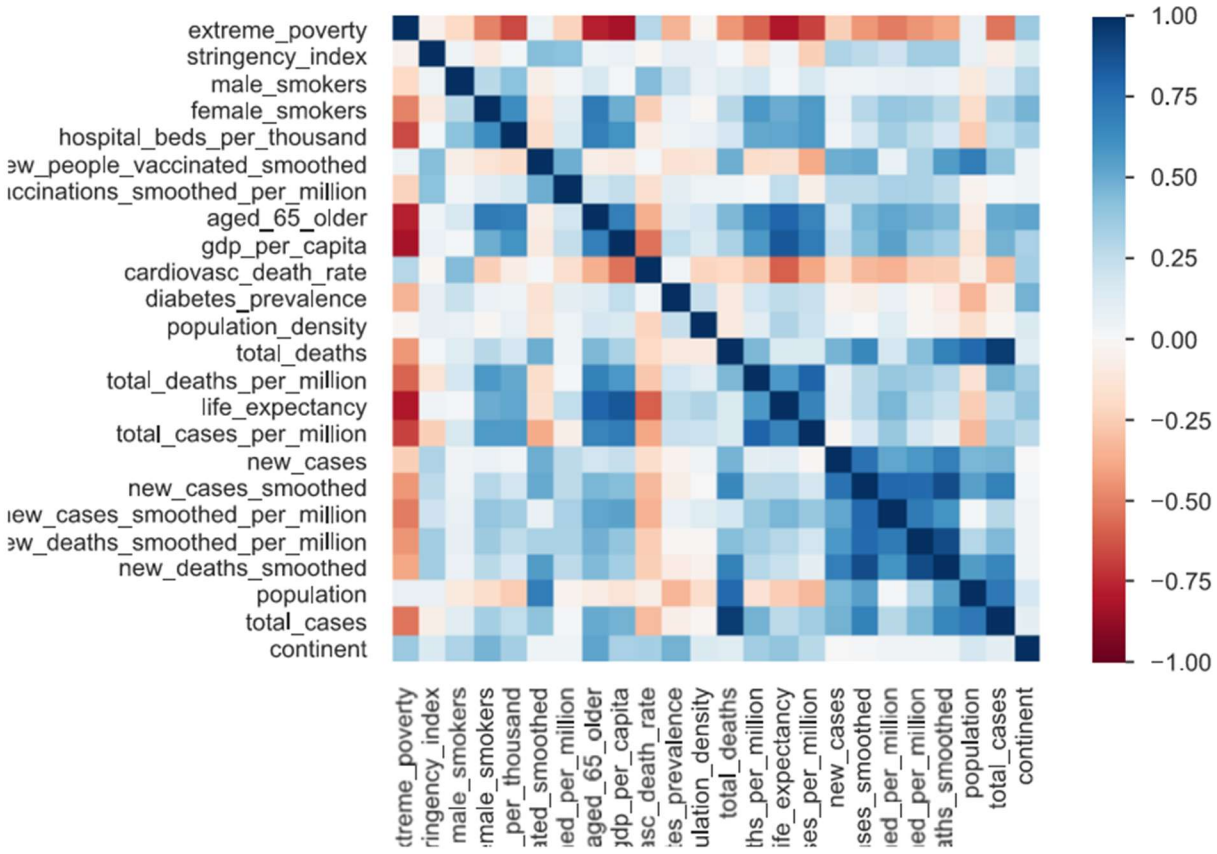

## total_cases

Real number (ℝ)

HIGH CORRELATION MISSING

| | |
|---|---|
| Distinct | 115953 |
| Distinct (%) | 48.3% |
| Missing | 14960 |
| Missing (%) | 5.9% |
| Infinite | 0 |
| Infinite (%) | 0.0% |
| Mean | 8362346.1 |

| | |
|---|---|
| Minimum | 1 |
| Maximum | $7.7115046 \times 10^8$ |
| Zeros | 0 |
| Zeros (%) | 0.0% |
| Negative | 0 |
| Negative (%) | 0.0% |
| Memory size | 12.0 MiB |




# Modeling Algorithms

The shape for my Training and Testing datasets are:

```
Training Features Shape: (28900, 17)

Training Prediction Shape: (12387, 17)

Testing Features Shape: (28900, 1)

Testing Prediction Shape: (12387, 1)
```

## Linear Regression

**Linear regression** is a method we can use to quantify the relationship between one or more predictor variables and a dependent variable or an outcome variable.

One of the most common reasons for fitting a regression model is to use the model to predict the values of new observations.

The steps to make predictions with a regression model are:

1. Collect the data.
2. Fit a regression model to the data.
3. Verify that the model fits the data well.
4. Use the fitted regression equation to predict the values of new observations.

**Examine each of the model's coefficients:**

Once the Linear Regression was trained I was able to examine each of the model's coefficients. Large coefficients on a specific variable mean that that variable has a large impact on the variable we're trying to predict. Similarly, small values have a small impact.

| Feature/Variable | Coeff |
|---|---|
| date | 0.012271 |
| new_cases | 14.761466 |
| total_deaths | 85.086849 |
| new_deaths | -2325.047296 |
| total_cases_per_million | 4.360587 |
| new_cases_smoothed_per_million | -24.709972 |
| total_deaths_per_million | -1204.878232 |
| new_deaths_smoothed_per_million | 24674.010368 |

| Feature/Variable | Coeff |
|---|---|
| new_vaccinations_smoothed | -8.907207 |
| new_vaccinations_smoothed_per_million | 7.372153 |
| population_density | 464.395830 |
| median_age | 45978.235161 |
| gdp_per_capita | 31.393894 |
| cardiovasc_death_rate | 4876.086804 |
| diabetes_prevalence | -130696.783992 |
| hospital_beds_per_thousand | 223064.224657 |
| life_expectancy | 15407.512203 |

### Making Predictions and Testing the Linear Regression Model

Intercept: [-23411752.86373093]

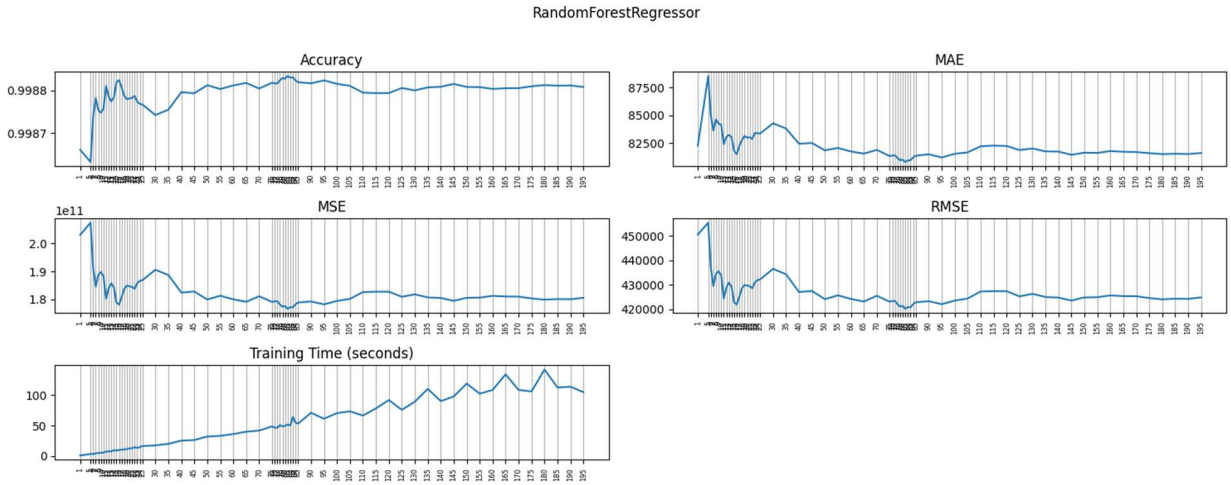
| | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| Linear Regression | 0.942278440 4304934 | 1407548.42 856267 | 8746630701 897.816 | 2957470.321 3891797 | 0.022003412246 7041 |

## Random Forest

A Random Forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the `max_samples` parameter if `bootstrap=True` (default), otherwise the whole dataset is used to build each tree.

From initial inspection I noticed that there are peaks in the measured accuracy as noted in the plots below, with the number of trees set around 15 and 80. Hence I ran estimators with the following number of trees:

- 5 to 25 in increments of 1
- 30 to 70 in increments of 5
- 75 to 85 in increments of 1
- 90 to 195 in increments of 5




| Comparison | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| RandomForestRegressor(n_estimators=1) | 0.9986601702214833 | 82270.22927262452 | 20302632783 1.2694 | 450584.4291930974 | 0.8929872512817383 |
| RandomForestRegressor(n_estimators=5) | 0.9986313866144769 | 88533.43192056188 | 20738794908 0.44373 | 455398.67048603 | 3.4839599132 53784 |
| RandomForestRegressor(n_estimators=6) | 0.9987379315412097 | 85011.87629504049 | 19124304353 3.13882 | 437313.4385462432 | 3.5795426368 71338 |
| RandomForestRegressor(n_estimators=7) | 0.9987823063897096 | 83617.06081260307 | 18451885910 0.5793 | 429556.5842826 5224 | 4.0489556789 39819 |

| Comparison | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| RandomForestRegressor(n_estimators=8) | 0.998754256421324 | 84629.94237910713 | 188769311037.38892 | 434475.90386279067 | 5.266948938369751 |
| RandomForestRegressor(n_estimators=9) | 0.998747626256070 7 | 84249.02990590493 | 189773989486.75705 | 435630.565372492 | 5.324516057968814 |
| RandomForestRegressor(n_estimators=10) | 0.998757421287808 2 | 84154.71599257286 | 188289734280.2123 | 433923.6502890944 | 5.714936733245585 |
| RandomForestRegressor(n_estimators=11) | 0.998811113540847 7 | 82395.21937221574 | 180153670175.35413 | 424445.132114098 | 7.479920148849487 |
| RandomForestRegressor(n_estimators=12) | 0.998786063801292 6 | 83055.96855574392 | 183949493134.76224 | 428893.33538161 | 7.891494750976562 5 |
| RandomForestRegressor(n_estimators=13) | 0.998774493625886 | 83237.86503219878 | 185702738407.29324 | 430932.40584492276 | 7.559918165206909 |
| RandomForestRegressor(n_estimators=14) | 0.998783780256146 3 | 83033.57517674059 | 184295522005.6969 | 429296.5432025943 | 9.623895406723022 |
| RandomForestRegressor(n_estimators=15) | 0.998819934556460 5 | 81801.66557412341 | 178817009029.05594 | 422867.60224573355 | 9.181899785995483 |
| RandomForestRegressor(n_estimators=16) | 0.998824654492339 | 81499.33344938242 | 178101790376.36057 | 422021.0781185705 | 10.08447289466857 9 |
| RandomForestRegressor(n_estimators=17) | 0.998807118415139 9 | 82246.56339900939 | 180759057303.40192 | 425157.68522208545 | 10.50488591194152 8 |
| RandomForestRegressor(n_estimators=18) | 0.998787627578955 7 | 82753.91316164796 | 183712531662.82303 | 428616.9988029208 | 11.16088008880615 2 |
| RandomForestRegressor(n_estimators=19) | 0.998779921988766 | 83145.27940157976 | 184880170795.09818 | 429976.9421667843 | 11.43387484550476 |

| Comparison | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| RandomForestRegressor(n_estimators=20) | 0.9987813174995488 | 82986.26834180998 | 184668707044.82983 | 429730.9705441648 | 12.799856901168823 |
| RandomForestRegressor(n_estimators=21) | 0.9987828048443164 | 83021.44202255052 | 184443327559.15558 | 429468.6572488795 5 | 12.699858188 62915 |
| RandomForestRegressor(n_estimators=22) | 0.9987874471275875 | 82879.81166839135 | 183739875717.4307 | 428648.8956213823 | 14.773845911026001 |
| RandomForestRegressor(n_estimators=23) | 0.9987740021751844 | 83415.01436990393 | 185777208636.9158 | 431018.8031 1294517 | 13.545848846435547 |
| RandomForestRegressor(n_estimators=24) | 0.9987685010917448 | 83385.52937555504 | 186610795699.79443 | 431984.7169747958 | 14.558843612670898 |
| RandomForestRegressor(n_estimators=25) | 0.9987671777461766 | 83351.87904093001 | 186811324151.58948 | 432216.7559819835 | 16.443820714 95056 |
| RandomForestRegressor(n_estimators=30) | 0.9987423054773966 | 84273.98493043784 | 190580254709.9377 | 436554.9847 4984535 | 17.571810245513916 |
| RandomForestRegressor(n_estimators=35) | 0.9987548487854693 | 83808.7984822798 | 188679549248.93 | 434372.5926 5396796 | 20.109785318374634 |
| RandomForestRegressor(n_estimators=40) | 0.9987962731814054 | 82441.1833232421 | 182402451124.6886 | 427085.9996823691 | 25.429721593 85681 |
| RandomForestRegressor(n_estimators=45) | 0.9987937894131337 | 82505.50820483839 | 182778820092.95108 | 427526.3969 5456356 | 26.366714954 37622 |
| RandomForestRegressor(n_estimators=50) | 0.9988127371702421 | 81844.36719786873 | 179907639284.7195 | 424155.2065986218 | 32.138652801 51367 |
| RandomForestRegressor(n_estimators=55) | 0.9988037855886993 | 82064.21889517605 | 181264085273.6556 | 425751.2011 41765 | 33.232645988464355 |

| Comparison | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| RandomForestRegressor(n_estimators=60) | 0.9988121707804315 | 81743.00311886116 | 17999346514 5.0096 | 424256.36724156496 | 36.223610162734985 |
| RandomForestRegressor(n_estimators=65) | 0.9988180850355254 | 81551.4621979619 | 17909726958 8.8007 | 423198.8534823797 | 40.115570068359375 |
| RandomForestRegressor(n_estimators=70) | 0.9988049026026444 | 81884.3170558996 | 18109482254 8.60403 | 425552.37344961904 | 41.900545597076416 |
| RandomForestRegressor(n_estimators=75) | 0.9988185463339694 | 81325.35768251662 | 17902736837 4.02917 | 423116.25869733386 | 48.929470300 67444 |
| RandomForestRegressor(n_estimators=76) | 0.9988168812661616 | 81380.78447056125 | 17927967848 6.8634 | 423414.3106779262 | 46.561502456 66504 |
| RandomForestRegressor(n_estimators=77) | 0.9988169035469272 | 81391.59413041951 | 17927630225 0.49323 | 423410.32374104107 | 46.672489404 678345 |
| RandomForestRegressor(n_estimators=78) | 0.9988242978925018 | 81150.85959328742 | 17815582646 1.11118 | 422085.0938627319 | 50.911448955 53589 |
| RandomForestRegressor(n_estimators=79) | 0.9988295123784766 | 80962.03258520315 | 17736566792 3.08737 | 421148.0356395924 | 48.878473758 69751 |
| RandomForestRegressor(n_estimators=80) | 0.9988281354485199 | 81004.05895898928 | 17757431609 4.12863 | 421395.6764065439 | 49.657460689 54468 |
| RandomForestRegressor(n_estimators=81) | 0.9988345145 80627 | 80797.44987 427081 | 17660767705 7.41666 | 420247.16186 71763 | 52.033439636 23047 |
| RandomForestRegressor(n_estimators=82) | 0.9988307409458331 | 80940.96626183628 | 17717950134 9.62244 | 420926.95488602587 | 50.270454645 15686 |
| RandomForestRegressor(n_estimators=83) | 0.9988312978 57623 | 80956.42509976939 | 17709511170 7.41144 | 420826.7003261692 | 64.421304464 34021 |

| Comparison | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| RandomForestRegressor(n_estimators=84) | 0.9988240208862749 | 81209.26644773514 | 1781978016111.94598 | 422134.8144988115 | 55.20939922332764 |
| RandomForestRegressor(n_estimators=85) | 0.9988197927555991 | 81359.03999259185 | 1788384963116.94797 | 422893.00812019577 | 53.33643102645874 |
| RandomForestRegressor(n_estimators=90) | 0.9988172699312318 | 81499.48456715373 | 1792207835111.3894 | 423344.75727400876 | 71.18022847175598 |
| RandomForestRegressor(n_estimators=95) | 0.9988241669787062 | 81202.36896321696 | 1781756640116.30707 | 422108.5926823891 | 61.44333410230615 |
| RandomForestRegressor(n_estimators=100) | 0.9988160304392695 | 81532.13136837007 | 1794086055056.0231 | 423566.5302004198 | 70.59323191642761 |
| RandomForestRegressor(n_estimators=105) | 0.9988111991808335 | 81660.51091889731 | 1801406930255.4936 | 424429.8446451 3516 | 73.57820200920105 |
| RandomForestRegressor(n_estimators=110) | 0.9987951170988314 | 82212.84323887948 | 1825776339755.09204 | 427291.04129982885 | 66.54828476905823 |
| RandomForestRegressor(n_estimators=115) | 0.9987942083528611 | 82271.4651019126 | 1827153375555.23032 | 427452.14650909207 | 78.32515406608582 |
| RandomForestRegressor(n_estimators=120) | 0.9987942498865728 | 82239.17462662469 | 1827090438923.3335 | 427444.7846123912 | 92.156005859375 |
| RandomForestRegressor(n_estimators=125) | 0.9988061890366128 | 81870.09755711634 | 1808998873664.4623 | 425323.2739510763 | 76.03417825698853 |
| RandomForestRegressor(n_estimators=130) | 0.9988004757087775 | 82011.18269028944 | 1817656361225.013 | 426339.8129720153 | 89.35403490066528 |
| RandomForestRegressor(n_estimators=135) | 0.9988076606129543 | 81751.64833980666 | 1806768972900.98596 | 425061.0512514478 | 110.41380882263184 |

| Comparison | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| RandomForestRegressor(n_estimators=140) | 0.9988090330841201 | 81731.88605450414 | 180468924766.9468 | 424816.34239627223 | 90.3770182132721 |
| RandomForestRegressor(n_estimators=145) | 0.9988158065988454 | 81448.08690200796 | 179442524366.50937 | 423606.5678982201 | 97.86094427108765 |
| RandomForestRegressor(n_estimators=150) | 0.9988086415 9277 | 81634.36264309355 | 180528248012.6586 | 424886.1588857168 | 119.17171311378479 |
| RandomForestRegressor(n_estimators=155) | 0.9988081890443534 | 81615.82210694355 | 180596823323.2549 | 424966.8496756599 | 102.61188960075378 |
| RandomForestRegressor(n_estimators=160) | 0.9988040347736479 | 81776.13230352386 | 181226325921.03717 | 425706.85444450757 | 108.69582533836365 |
| RandomForestRegressor(n_estimators=165) | 0.9988056243588128 | 81716.51331430068 | 180985453801.32126 | 425423.85194217926 | 134.41155004501343 |
| RandomForestRegressor(n_estimators=170) | 0.9988058618493673 | 81693.57210073179 | 180949466516.9991 | 425381.55403942836 | 108.73582243919373 |
| RandomForestRegressor(n_estimators=175) | 0.9988101197478612 | 81592.81173718991 | 180304261051.83047 | 424622.4923998144 | 106.28985452651978 |
| RandomForestRegressor(n_estimators=180) | 0.9988129997401566 | 81506.8016316389 | 179867851689.02185 | 424108.3018393083 | 142.30246543884277 |
| RandomForestRegressor(n_estimators=185) | 0.9988115696300917 | 81537.38752964638 | 180084558318.13327 | 424363.7099448223 | 112.79077553749084 |
| RandomForestRegressor(n_estimators=190) | 0.9988120278727799 | 81514.88231932459 | 180015120146.39786 | 424281.8876011535 | 114.04177045822144 |
| RandomForestRegressor(n_estimators=195) | 0.9988085740942615 | 81601.85500431593 | 180538476158.46445 | 424898.19505202 | 105.09187006950378 |

From the results above, the optimal estimators are found with `n_estimators=16` and `n_estimators=81`. The latter shows smaller `MAE`, `MSE` and `RMSE`. I will proceed then with `n_estimators=81`.

## Logistic Regression

Logistic regressions, also referred to as Logit Models, are powerful alternatives to linear regressions that allow to model a dichotomous, binary outcome (i.e., 0 or 1) and provide notably accurate predictions on the probability of said outcome occurring given an observation. The parameter estimates within Logit Models can provide insights into how different explanatory variables or features contribute to the model predictions.

### Making Predictions and Testing the Logistic Regression Model

Intercept: [ 1.73814779e-18 -1.24853695e-19 1.15253590e-18 ... 3.71401863e-19 3.69880204e-19 -3.92605189e-19]

| | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| Logistic Regression | 0.012997497376281586 | 2175357.539032857 | 129518516.83426173 | 11380.620230649194 | 930.3401651382446 |

# Results

| Model | Accuracy | MAE | MSE | RMSE | Training Time (seconds) |
|---|---|---|---|---|---|
| Linear Regression | 0.9422784404304934 | 1407548.428562267 | 8746630701897.816 | 2957470.3213891797 | 0.022003412246 7041 |
| Random Forest | 0.9988345145806 27 | 80797.44987427081 | 176607677057.41666 | 420247.1618671763 | 52.03343963623047 |

| | | | | | |
|---|---|---|---|---|---|
| Logistic Regression | 0.012997497376281586 | 2175357.539032857 | 129518516.83426173 | 11380.620230649194 | 930.3401651382446 |

As expected, the least accurate model is Logistic Regression as this model is used for classification and not prediction.

Both Linear regression and Random Forest show good accuracy. For Random Forest the best accuracy was calculated with 81 decision trees.

The Mean Absolute Error (MAE) represents the average of the absolute difference between the actual and predicted values in the dataset. It measures the average of the residuals in the dataset. This means that the smaller the difference the better the prediction is.

Similar results are obtained for Mean Squared Error (MSE) which represents the average of the squared difference between the original and predicted values in the dataset. It measures the variance of the residuals. as well as with Root Mean Squared Error (RMSE) which is the square root of Mean Squared error. It measures the standard deviation of residuals.

This means that the best model for this use case is Random Forest configured with 81 decision trees.

# Conclusion

In this document, various prediction techniques are proposed and evaluated for predicting the number of COVID-19 cases based on a publicly available dataset of observations.

Due to the large size of the dataset both in terms of the number of features and the number of observations a selection of the data is recommended for efficient processing and manipulation. The original dataset includes 346,567 observations and 67 features. After One Hot Encoding, a total of 552 columns remained. However:

- Even with this reduction the processing of the file was slow, with the Backwards Elimination algorithm run taking 90 minutes to complete.
- Additional compute resources were required to prevent the Jupyter Kernel from crashing, but the dataset was manageable only after restricting the observations to only those from North America.

Performance significantly improved using only data from North America, but it would have been great to run the models against the whole dataset.

It was also interesting to see the impact on how the split is performed. Based on some of the articles, some models were run by selecting training and testing dataset based on the observation date. At that moment my approach was to split the dataset selecting observations from 2021 and 2022 as the training dataset and observations from 2023 as the testing dataset. But this ended up affecting the models and yielding low accuracy. The best approach was to use a random split with 70% for training dataset and 30% for testing dataset.

It is interesting to see that attributes like `testing` or `ICU` don't add value in the prediction of the number of cases and that derived *weekly* columns are not relevant.

On the other hand, it is also interesting to see how information like gross domestic product, purchasing power, cardiovascular and diabetes are highly relevant in how they influence the number of cases.

The best models to predict the number of COVID-19 cases are Linear Regression and Random Forest. Random Forest has a higher accuracy and smaller `MAE`, `MSE` and `RMSE`.

# GitHuB Repository

https://github.com/aamadorc/CIND820

https://aamadorc.github.io/CIND820/changes

# References

Ahouz, F., & Golabpour, A. (2021, June 7). Predicting the incidence of COVID-19 using data mining. *BMC Public Health, 21*(1), 1087.

Barua, A., Hridoy, M., Uddin, K., Chowdhury, R., & Ahamed, J. (n.d.). Analysis and Prediction of the Spread of COVID-19 in Bangladesh Using Statistical and Machine Learning Approach. Retrieved from https://ssrn.com/abstract=4592228

Brownlee, J. (2020). *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python.* Machine Learning Mastery.

Gangloff, C., Rafi, S., Bouzillé, G., Soulat, L., & Cuggia, M. (2021). Machine learning is the key to diagnose COVID-19: a proof-of-concept study. *Scientific Reports, 11*(1), 7166.

Gothai, E., Thamilselvan, R., Rajalaxmi, R., Sadana, R., Ragavi, A., & Sakthivel, R. (2023). Prediction of COVID-19 growth and trend using machine learning approach. *Materials Today: Proceedings, 81*(2), 597-601.

Heredia Cacha, I., Sáinz-Pardo Díaz, J., Castrillo, M., & López García, Á. (2023). Forecasting COVID-19 spreading through an ensemble of classical and machine learning models: Spain's case study. *Scientific Reports, 13*(1).

Iwendi, C., Bashir, A., Peshkar, A., Sujatha, R., Chatterjee, J., Pasupuleti, S., . . . Jo, O. (2020, July 3). COVID-19 patient health prediction using boosted random forest algorithm. *Frontiers in Public Health, 8*, 357.

Koehrsen, W. (2018, 1). *Random Forest in Python. A Practical End-to-End Machine Learning.* Retrieved November 26, 2023, from Towards Data Science: https://towardsdatascience.com/random-forest-in-python-24d0893d51c0

Kuhn, M., & Johnson, K. (2020). *Feature Engineering and Selection: A Practical Approach for Predictive Models.* CRC Press, Taylor & Francis Group.

McCullum, N. (n.d.). *Linear Regression in Python - A Step-by-Step Guide.* Retrieved November 26, 2023, from https://www.nickmccullum.com/python-machine-learning/linear-regression-python

Meraihi, Y., Gabis, A., Mirjalili, S., Ramdane-Cherif, A., & Alsaadi, F. (2022, 5 12). Machine Learning-Based Research for COVID-19 Detection, Diagnosis, and Prediction: A Survey. *SN computer science, 3*(4).

Painuli, D., Mishra, D., Bhardwaj, S., & Aggarwal, M. (2021). Forecast and prediction of COVID-19 using machine learning. *Data Science for COVID-19*, 381–397.

Saqib, M. (2020, September 10). Forecasting COVID-19 Outbreak Progression Using Hybrid Polynomial-Bayesian Ridge Regression Model.

Silipo, R., Adae, I., Hart, A., & Berthold, M. (2014). Seven Techniques for Dimensionality Reduction. Retrieved November 26, 2023, from KNIME: https://www.knime.com/sites/default/files/inline-images/knime_seventechniquesdatadimreduction.pdf

Singh, H. (2021, April 5). *Missing Value Ratio Implementation in Python.* Retrieved November 26, 2023, from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-missing-value-ratio-and-its-implementation/

Singh, K., Kumar, S., Dixit, P., & Bajpai, M. (2021). Kalman filter based short term prediction model for COVID-19 spread. *Applied Intelligence, 51*(5), 2714–2726.

Tuli, S., Tuli, S., Tuli, R., & Gill, S. (2020, September). Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing. *Internet of Things (Netherlands), 11*.

Vishal, R. (2022, 4). *Feature Selection - Correlation and P-value.* Retrieved November 26, 2023, from Kaggle: https://www.kaggle.com/code/bbloggsbott/feature-selection-correlation-and-p-value

Wang, L., Shen, H., Enfield, K., & Rheuban, K. (2021). COVID-19 Infection Detection Using Machine Learning. In *2021 IEEE International Conference on Big Data (Big Data)* (pp. 4780-4789). IEEE.

Zoabi, Y., Deri-Rozov, S., & Shomron, N. (2021). Machine learning-based prediction of COVID-19 diagnosis based on symptoms. *npj Digital Medicine, 4*(1).