MSC DATA SCIENCE & ANALYTICS DISSERTATION PROJECT

Department of Computer Science

Modelling The Spread Of Covid-19 Through Novel Machine Learning Techniques For Future Use Cases

NASSIM BENATIA
STUDENT ID # 1238355
Academic Year 2019 – 2020

Brunel University
Department of Computer Science
Uxbridge, Middlesex UB8 3PH
United Kingdom
Tel: +44 (0) 1895 203397
Fax: +44 (0) 1895 251686

A report submitted in partial fulfilment of the requirement for the degree of Master of Science

Abstract

The aim of this project is to employ machine learning techniques to model the spread of Covid-19. A successful model will allow for the prevention of future spreads of the disease, as it becomes easier to identify and prioritise riskier areas for disease control. This project employs the Android Global Mobility Index as the main input and uses three separate machine learning algorithms to model how peoples movements have an impact on the spread rate of the disease in each region. The result of the research project showed that Covid-19 can indeed be modelled successfully and lends evidence to the idea that regional lockdowns can help tackle the disease.

Acknowledgments

I would like to acknowledge my supervisor Dr Crina Grosan for guiding me towards the successful completion of this dissertation project.

TOTAL NUMBER OF WORDS: 7008

I certify that the work presented in the dissertation is my own unless referenced

Signature: Nassim Benatia

Date 28/09/2020

Contents

Project Context

With the onset of the Covid-19 outbreak in late 2019, we have witnessed the world brought to its knees by a virus with which just six months prior, most of the world new nothing about. In late 2019 or even early 2020, most people would have been sceptical of the idea of a what can be considered a world scale lockdown. The death toll of the disease at the time of writing exceeds 926,000 with cases rampant at over 29 million {Worldometers 2020}. World superpowers are at loggerheads in finding an effective vaccine with some claiming to have succeeded in that regard, i.e. Russia, with the head of state Vladamir Putin claiming his daughter to have already been vaccinated {Walsh, Fergus 2020}. The world economy has taken a hefty blow as a result of the pandemic, with a 5.2 percent contraction in global GDP in 2020, and with the World Bank claiming "the deepest global recession in decades" despite the extraordinary efforts of governments to counter the downturn with fiscal and monetary policy support.

With the speed at which this virus has spread, it has become important to meet the virus's remarkable spread with an equally remarkable response. The world must leverage all available resources in combatting the disease. As widely quoted, "in any battle, information is king" and with this in mind, there is a large focus on using technology to align our interests in the fight against the virus.

What is Covid-19?

In late December 2019, a cluster of patients were admitted to hospital with an initial diagnosis of pneumonia of an unknown ethology. These patients were epidemiologically linked to a seafood and wet animal wholesale market in Wuhan, Hubei Province, China {Rothan, Hussin A 2020}. This new virus was quickly categorized as a type of coronavirus. Coronaviruses generally are major pathogens that primarily target the human respiratory system. Previous outbreaks of coronaviruses (CoVs) include the severe acute respiratory syndrome (SARS)-CoV and the Middle East respiratory syndrome (MERS)-CoV which have been previously characterized as agents that are a great public health threat. {Rothan, Hussin A 2020}. The International Committee on Taxonomy of Viruses (ICTV) announced "severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)" as the name of the new virus on 11 February 2020. This name was chosen because the virus is genetically related to the coronavirus responsible for the SARS outbreak of 2003. The WHO also begun referring to the virus as COVID-19 when communicating with the public {WHO 2020}.

Disease Spread:

Person-to-person transmission of the virus occurs primarily via direct contact or through droplets spread by coughing or sneezing {Rothan, Hussin A 2020}. The latest guidelines from Chinese health authorities described three main transmission routes for COVID-19: 1) droplets transmission, 2) contact transmission, and 3) aerosol transmission. {Adhikari, Sasmita Poudel 2020}. The symptoms of COVID-19 infection appear after an incubation period of approximately 5.2 days {Rothan, Hussin A 2020} and the most commonly reported symptoms are fever, cough, myalgia or fatigue, pneumonia, and complicated dyspnea, whereas less common reported symptoms include headache, diarrhea, hemoptysis, runny nose, and phlegm producing coughs {Adhikari, Sasmita Poudel 2020}. By January 2, 2020, 41 admitted hospital patients had been identified as having laboratory-confirmed COVID-19 infection, less than half of these patients had underlying diseases, including diabetes, hypertension, and cardiovascular disease {Rothan, Hussin A 2020}. Early reports predicted the onset of a potential Coronavirus outbreak given the estimate of a reproduction number for COVID-19, which was deemed

to be significantly larger than 1 (ranges from 2.24 to 3.58) {Rothan, Hussin A 2020}. At the time of writing, world cases have surpassed 29 million and deaths over 926 thousand {Worldometers 2020}.

Efforts At Containment:

Upon realizing the extent of the Virus spread in Wuhan, Chinese authorities began preventing travel to and from the city. Between the 23rd and 25th January, travel restrictions were imposed on 18 additional cities affecting nearly 60 million people {Peeri, Noah C 2020;}. On January 30, 2020, the WHO announced a state of international public health emergency, later declaring COVID-19 to be a pandemic in March 2020. {48 Surveillances, Vital 2020; 44 Khachfe, Hussein H 2020;}. Since then, many national level lockdowns have taken place, with the UK imposing a full scale lockdown for over seven weeks from 23rd March 2020, and with a second full scale lockdown potentially on the horizon.

Use of Technology In The Fight Against Covid-19:

"digitalized economies have lower epidemic risks" {World Bank 2020}.

With government policy unconvincing in its efforts to curb the disease, and recent lockdowns being described as late and in vain, the demand for effective statistical analysis and modelling has become more important, as the disease has penetrated deep into society, and now requires a more sensitive approach to its management.

It is therefore natural that Big Data and Artificial intelligence have been at the forefront of Covid-19 preparedness with companies such as Apple and Google collecting and publicising different repositories of tracking data to help inform our understanding of the epidemiology of the disease. Google's Android Global Mobility Index provides many indicis of varying categories for different world regions, which together track peoples changes in behaviour during the pandemic. Apples Mobility Trends are of a similar nature. The UK Government also publicised mobility data for London specifically which is again like Googles Android report although it has additional indicis for public transport usage.

Industry 4.0 also entered the arena early with IoT technologies allowing hospital staff to update live databases with their current status in terms of their available resources and case counts, and these databases have fed through to inform dashboards which inform daily policy decisions. Such examples are the Johns Hopkins University (MD, USA) coronavirus dashboard and the web-based platform HealthMap. In the UK, the government launched an automated chatbot service on WhatsApp that allows the public to get answers to the most common questions about COVID-19 directly from the government {JIANG, NAN 2020}. Educators have embraced remote teaching via applications such as Microsoft teams and Zoom, and Artificial Intelligence has been used to help diagnose the disease through downloadable detection software.

Research Aim & Objectives:

This research project employs machine learning techniques to model the spread of Covid-19. A successful model will allow for the prevention of future spreads of the disease, as it becomes easier to identify and prioritise riskier areas for disease control. This particular model employs the Android Global Mobility Indices to see how people's movements have an impact on the spread rate of the disease in each region. The research project therefore has two aims: The first is, as mentioned above, to model the spread of the disease to allow us to intervene earlier in future scenarios. The second aim of the project is to see whether regional lockdowns prove an effective means to curb the disease, as we analyse movements through googles mobility indicis. If there is a strong enough relationship

between the movement of people and the spread of the disease, then we should be able to see this in the data and create a suitable model for future use cases.

Methodology

What is Machine Learning?

Machine learning is a field of artificial intelligence that employs statistical methods to help computers learn independently of human intervention. The computer is taught to make accurate predictions through exposure to new data or environments. With the advent of increasingly efficient processing power and cloud computing, machine learning has become part of daily life. This can be seen through the onset of industry 4.0, an expression used in reference to the automation of traditional manufacturing processes using modern smart technology.

Unsupervised Vs Supervised Machine Learning:

Machine learning falls under two main categories which are unsupervised and supervised learning. Unsupervised machine learning is the process of learning the intrinsic structure of data with no prior exposure to examples. A common algorithm in this domain is cluster analysis, with notable examples including K-means clustering and hierarchical clustering. The aim of cluster analysis is to "create groups of objects, or clusters in such a way that objects in one cluster are very similar and objects in different clusters are distinct." {Gan, Guojun 2007}. Another useful unsupervised method is dimensionality reduction. The most common dimensionality reduction technique is principal component analysis, and these methods generally involve ridding a dataset of noise to make it easier to spot potential patterns that would otherwise be missed.

Supervised learning on the other hand is the process of taking existing examples of an output of some sort and attempting to replicate it accurately when exposed to new inputs. There are several supervised learning algorithms used in a variety of contexts that employ one or several different statistical tools. Some common supervised methods are listed and summarised below:

Linear Regression:

Linear regression is one of the earliest forms of supervised learning that involves creating a coefficient for use with an independent variable to predict an output. This is done using a method called least squares, which involves minimising a combined error term. When carried out effectively, linear regression is believed to be the best unbiased linear estimator of a dependent variable. Linear regression analysis is however sensitive to issues of multicollinearity, autocorrelation, and outliers.

Decision Trees & Random Forests:

A decision tree is a machine learning algorithm that involves creating splits within a feature space to minimize information entropy. In decision tree regression, for any given instance of data within the "leaf" of a "tree", the average Y value is taken and is used when making predictions. Decision trees can be applied to both regression and classification problems.

A random forest is an extension of a decision tree. It is a type of ensemble method that takes advantage of bootstrap sampling with replacement. Many instances of smaller trees are then created. This helps remedy the issue of potential overfitting in the training of models and helps make dominant features less problematic. Random forests are also efficient in training as they can handle

large data sets due to its sampling nature. Decision trees and random forests unlike linear regression do not suffer from issues of multicollinearity and outliers.

Support Vector Regressions (SVR's):

SVR's are a particular group of supervised learning algorithms that in the simplest terms split and categorise datapoints within a dataspace using a hyperplane which is as wide as possible. New datapoints are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. Support Vector Regressions have many advantages, some of which are that they are not prone to overfitting and they are also insensitive to noise. They can also be used for both regression and classification applications. Support Vector Regressions in simplest terms perform linear regressions, however they can perform non-linear predictions using kernel functions. Some notable kernels are the polynomial kernel used in image processing, the gaussian kernel used as a general-purpose kernel for unexplored data, and the radial basis function kernel (RBF) used often in regression applications.

Artificial Neural Networks (ANN's):

An artificial neural network is a network of artificial neurons, inspired by the human brain, that attempts to simulate the way the brain processes information and learns from exposure to new experiences. A key characteristic of ANN's are that they perform better when acquiring new data/information and thus have the ability to learn. ANN's have been used in a variety of applications in multiple disciplines, a few of which are Sales forecasting, Industrial process control, Customer research, Risk management & Target marketing. Some notable uses of ANN's in practise are Google's cloud speech API, Mastercard's use of ANN's in tackling fraud prevention, and NASA's use of ANN's to help optimize dynamics of new materials.

The Neuron:

A neuron or nerve cell is an electrically excitable cell that communicates with other cells via specialized connections called synapses. {Various 2020}. Neurons receive signals via the dendrites which are a short branched extension of a nerve cell, along which impulses received from other cells at synapses are transmitted to the cell body. Neurons are highly specialized for the processing and transmission of cellular signals. Given their diversity of functions performed in different parts of the nervous system, there is a wide variety in their shape, size, and electrochemical properties. {Various 2020;}

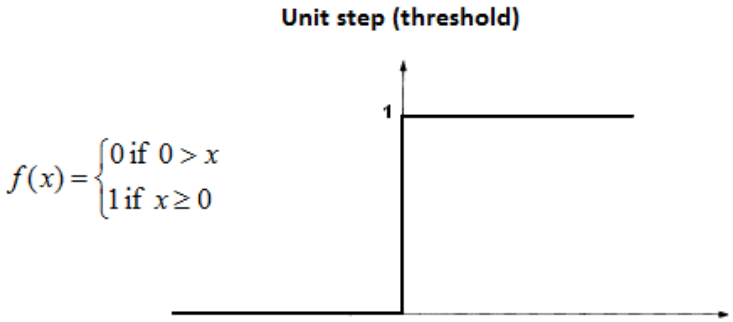
An artificial neuron is a mathematical function conceived as a model of biological neurons. The neuron in an artificial neural network receives input signals/data. These signals/data are given weights based on their importance through a process of backpropagation which will be touched on further down. Input signals can also come from other neurons that feed forward to create a more powerful application. These are called hidden layers. The weighted sum of inputs are usually passed through a non-linear function known as an activation function which determines the output of a neuron.

Activation Functions:

Activation functions were introduced to ANN's to avoid the problem of non-linearly separable data. Data of this type cannot be separated by a straight line and therefore a transformation needs to take place to enable the ANN to deal with this type of data characteristic. There are three main types of activation function used for different applications:

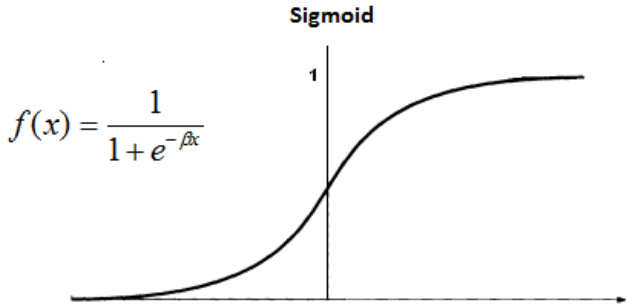
The Threshold Function:

This function is used in binary problems to output a 0 or 1 (corresponding to no and yes respectively). If the input value is less than 0, the output will be a zero and if equal to or more than 0, the function will pass on a 1.



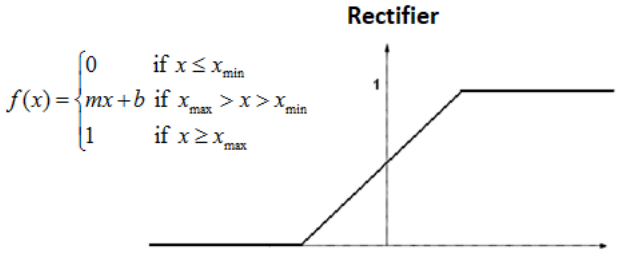
The Sigmoid Function:

The sigmoid function is similar to the threshold function in that is it used for binary classification although instead of outputting a hard yes/no, it outputs a probability of yes instead which in some use cases can be more useful (cite examples if possible.)



The Rectifier Function:

This function is mainly implemented in hidden layers of Artificial Neural Networks so as to make the network sparse and computationally efficient. It also has the ability to avoid the boundaries of learning associated with other activation functions such as the sigmoid, which suffers from the vanishing gradient problem.



There are other types of activation functions although many of those are extensions of the above types.

The Learning Mechanism of ANN's:

Artificial Neural Networks learn through a process called backpropagation. This process involves updating the weights of the input variables to minimise the difference or error between the predicted and actual values for the dependent variable. This process is made computationally feasible through a mathematical process termed gradient descent.

Stochastic gradient descent is better than batch gradient descent as it avoids local minimums. Stochastic gradient descent is achieved by feeding one instance of data at a time into a neural network and updating the weights. It is also a faster method. There is also mini batch gradient descent which inputs a few rows at a time, and this can be experimented with to achieve optimal results.

Data Overview

Data Sourcing:

With open source data widely available in the wake of the corona virus pandemic, there were many instances of datasets that were available for analysis. The final dataset used for analysis is a combination of two separate datasets which were merged and cleaned.

The first of two datasets was the "Country Level Cases Dataset" which includes daily cases for each country ("new_cases") for a rough period of 6 months between January and July 2020. The dataset was sourced from the Databiology web page which is an open source web page that brings together data scientists and provides a cloud-based computing platform with a simplified API for modelling data. The new_cases variable will act as the dependent variable which we will eventually use in a regression model. There are also other experimental variables provided by this dataset of which some examples are "stringency Index", "Population Density" and "Median Age". The second of the two datasets is Googles Android Global Mobility Report "Global_Mobility_Report" which comprises of mobility indicis that will be part of the independent variables in the analysis. The variable breakdown for both datasets is provided below:

Country Level Cases Dataset:

iso_code	new_tests_smoothed_per_thousand
continent	tests_units
region	stringency_index
date	population
total_cases	population_density
new_cases	median_age
total_deaths	aged_65_older
new_deaths	aged_70_older
total_cases_per_million	gdp_per_capita
new_cases_per_million	extreme_poverty
total_deaths_per_million	cvd_death_rate
new_deaths_per_million	diabetes_prevalence
total_tests	female_smokers
new_tests	male_smokers
total_tests_per_thousand	handwashing_facilities
new_tests_per_thousand	hospital_beds_per_thousand
new_tests_smoothed	life_expectancy

Global Mobility Report:

country_region_code
country_region
sub_region_1
sub_region_2
iso_3166_2_code
census_fips_code
date
retail_and_recreation_percent_change_from_baseline
grocery_and_pharmacy_percent_change_from_baseline
parks_percent_change_from_baseline
transit_stations_percent_change_from_baseline
workplaces_percent_change_from_baseline
residential_percent_change_from_baseline

Data Merging:

To combine the two datasets, an inner merge was completed on the country/region and date columns. As the Android Global Mobility Report had subregion columns which were absent in the country level cases dataset, sub regional data was deleted to avoid duplicated rows. It would have been beneficial to have sub regional data included in the analysis, however this was not possible as the Country Level Cases dataset was lacking this level of granularity.

The next step was to output the percentage levels of Null values in each variable to see whether it was worth keeping all the variables. Any variable with over 20% missing values was considered void and deleted. The newly merged dataset was also cleared of any variables which were not of interest or considered not useful for our use purposes. The variable breakdown of the final dataset can be seen below alongside summary statistics for each variable.

Summary Statistics Final Dataset:

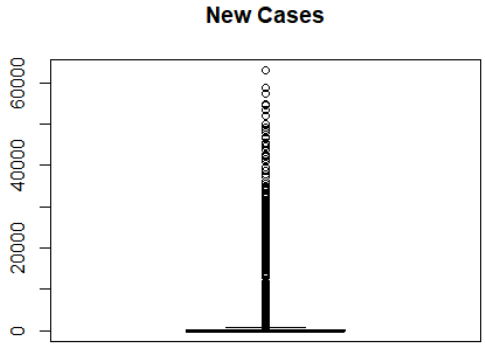
	Mean	Std	Min	25%	50%	75%	Max
Total Cases	32417.75	156024.16	0.00	114.00	1218.00	11071.50	3118008.00
New Cases	717.89	3277.88	0.00	1.00	25.00	263.00	63004.00
Stringency Index	64.58	25.86	0.00	50.00	72.22	84.26	100.00
Population	46776499.33	137729913.97	38137.00	5094114.00	11589616.00	37846605.00	1380004385.00
Population Density	250.30	799.16	1.98	35.61	88.13	227.32	7915.73
Median Age	32.60	9.00	15.10	25.30	31.90	41.20	48.20
GDP Per Capita	23023.68	21173.51	926.00	6426.67	16745.02	35220.08	116935.60
retail_and_recreation_percent_change_from_baseline	-35.09	26.83	-97.00	-57.00	-33.00	-13.00	42.00
grocery_and_pharmacy_percent_change_from_baseline	-16.33	22.60	-97.00	-29.00	-12.00	0.00	94.00
parks_percent_change_from_baseline	-11.48	46.83	-95.00	-41.00	-17.00	4.00	517.00
transit_stations_percent_change_from_baseline	-39.37	25.20	-95.00	-59.00	-41.00	-20.00	39.00
workplaces_percent_change_from_baseline	-28.05	23.47	-92.00	-45.00	-27.00	-9.00	80.00
residential_percent_change_from_baseline	13.68	10.35	-13.00	5.00	13.00	20.00	55.00

The main thing to take from the above summary statistics is that our dependent variable (new_cases) is an increasing function and rightward skewed, as the mean is significantly larger than the median,

(717.89 compared to 25). This is also the case with total_cases. All other variables seem to be normally distributed with similar mean and median figures.

To ascertain how to deal with the remaining missing values in the dataset, I had to look at the spread of the data. Looking at the spread of the dependent variable (new_cases), I realised that the data was increasing, as the mean was significantly higher than the median and many datapoints lied above the whisker of the extremity Q3 + IQR. For this reason, I decided it was inappropriate to replace null values with a measure of central tendency (i.e. the mean or median figure) as a mean figure would have been highly biased upward and the median figure would have been too much of a generalization (considering the exponential nature of the spread).

Boxplot: new_cases



For the mobility indicis, I could see a trend in the data, in that rows with missing data would tend to have missing data across a multitude of variables suggesting that for a particular country and day, there was no data collection. Due to this fact and due to the fact that for a multitude of variables the null values were less than 3%, I thought it was appropriate to completely delete rows with missing data. It was also necessary to do this as the Machine learning algorithms used later on in the analysis would not run with null values present in the dataset. The majority of the data was preserved with a minimum of over 75% of the data remaining after the deletion (over 12,000 rows of full data).

Exploratory Analysis

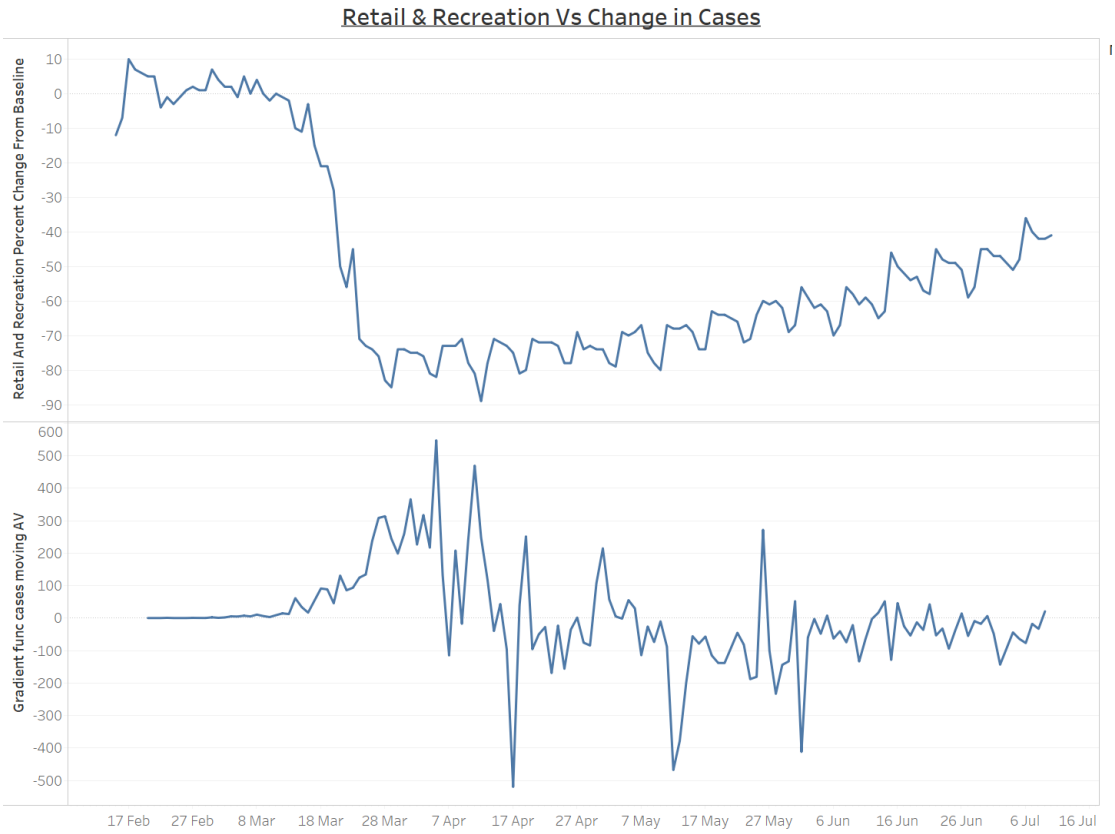
The goal of any exploratory analysis is to decipher the inherent characteristics of a dataset. By summarizing the data in different ways, we can tease out important patterns in the data which will inform our model later.

Time Series Plots:

As the aim of this research is to model the spread of Covid-19, it makes sense to begin by plotting input variables against our dependent variable new_cases on a time series graph and looking for a relationship.

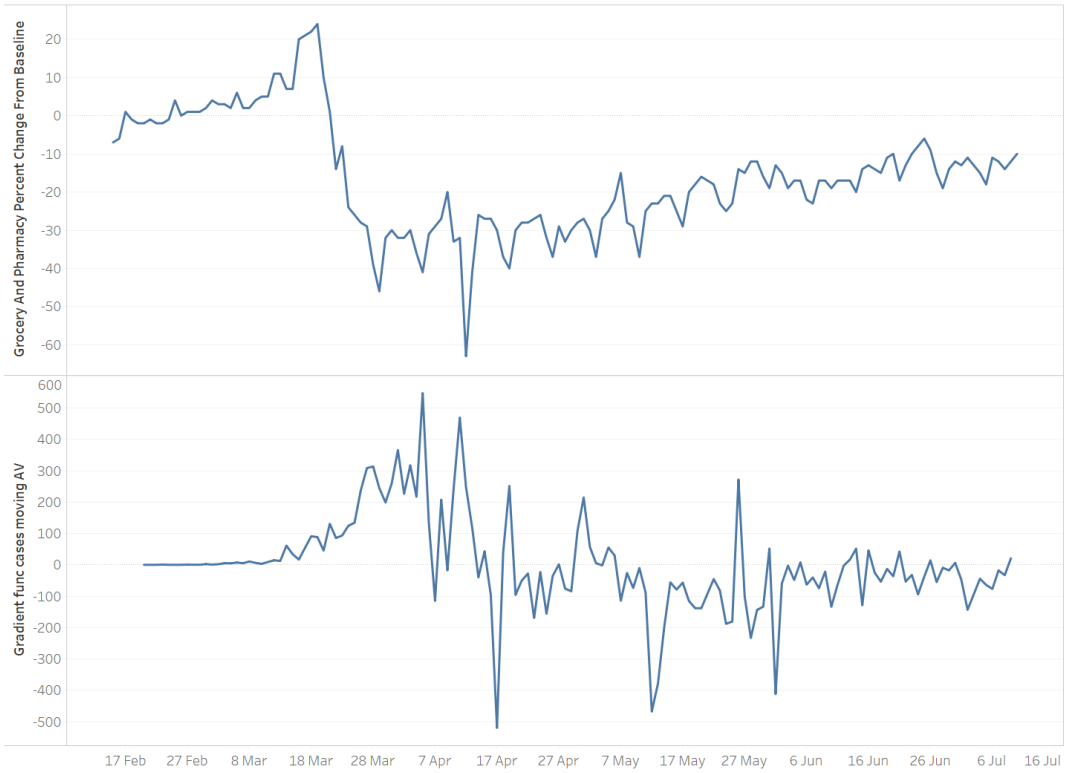
It must be noted that the time series analysis was completed on UK data exclusively, as plotting all country data was too noisy for useful analysis. After manipulating the dependent variable (new_cases)

by transforming it to show percentage change over time and adding a 6 day moving average, the data began to reveal some trends:

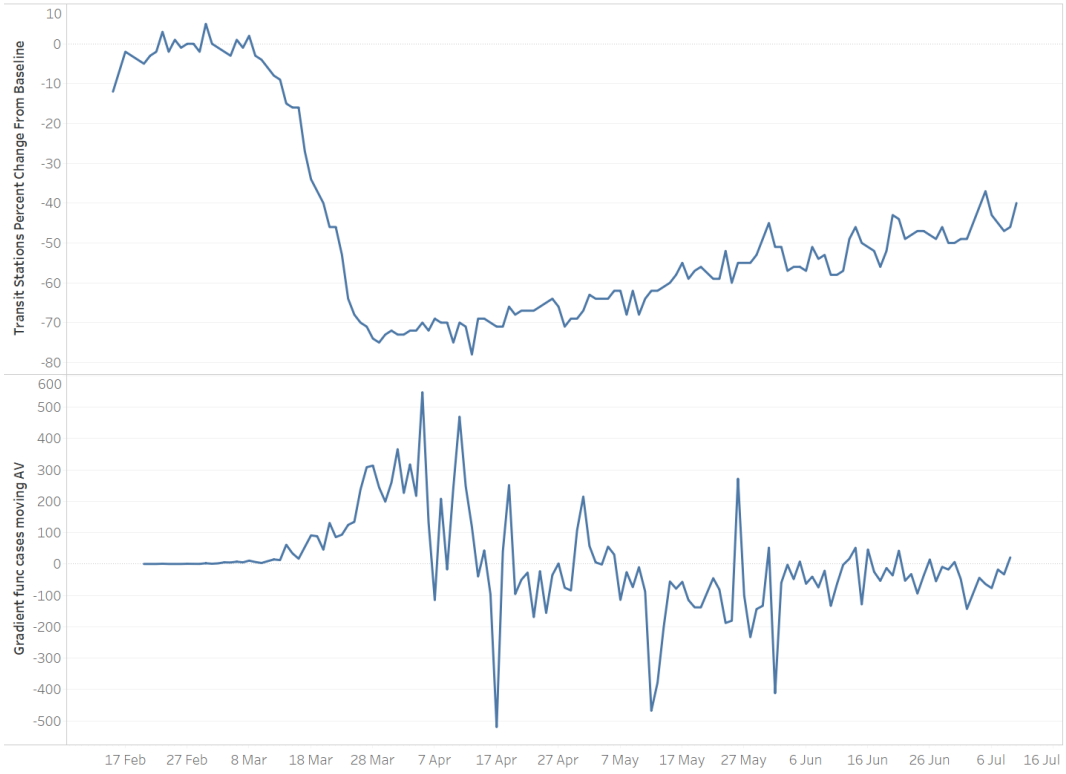


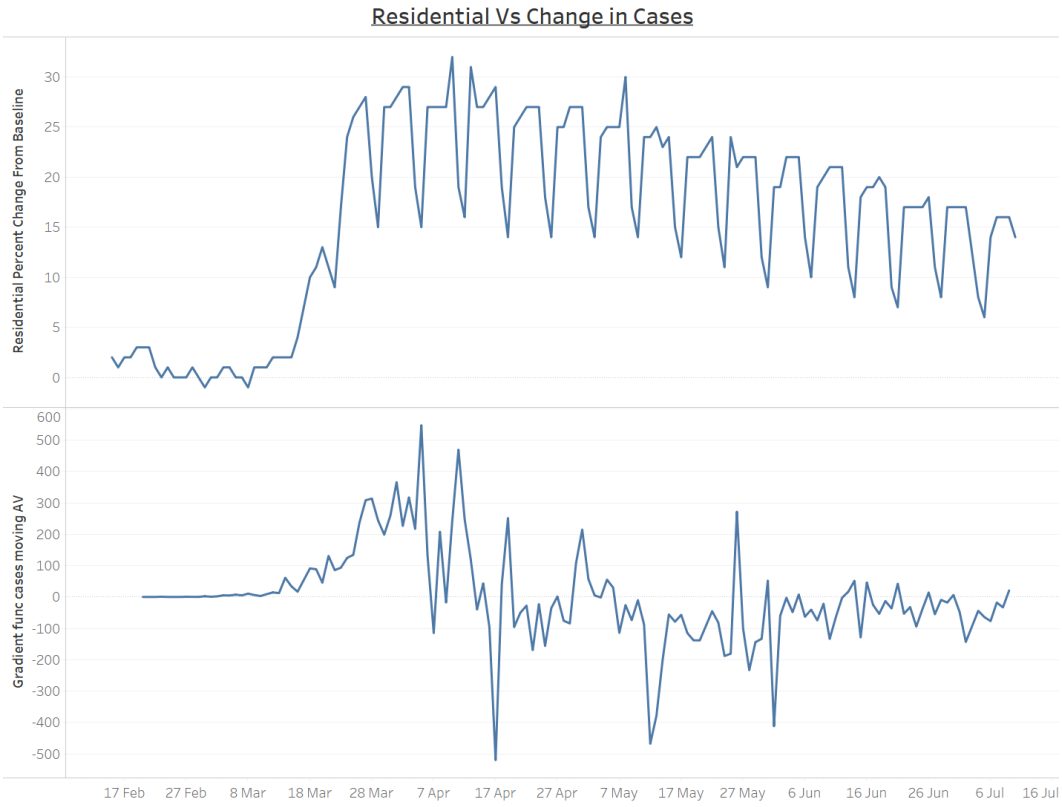
Graphs Continue On Next Page

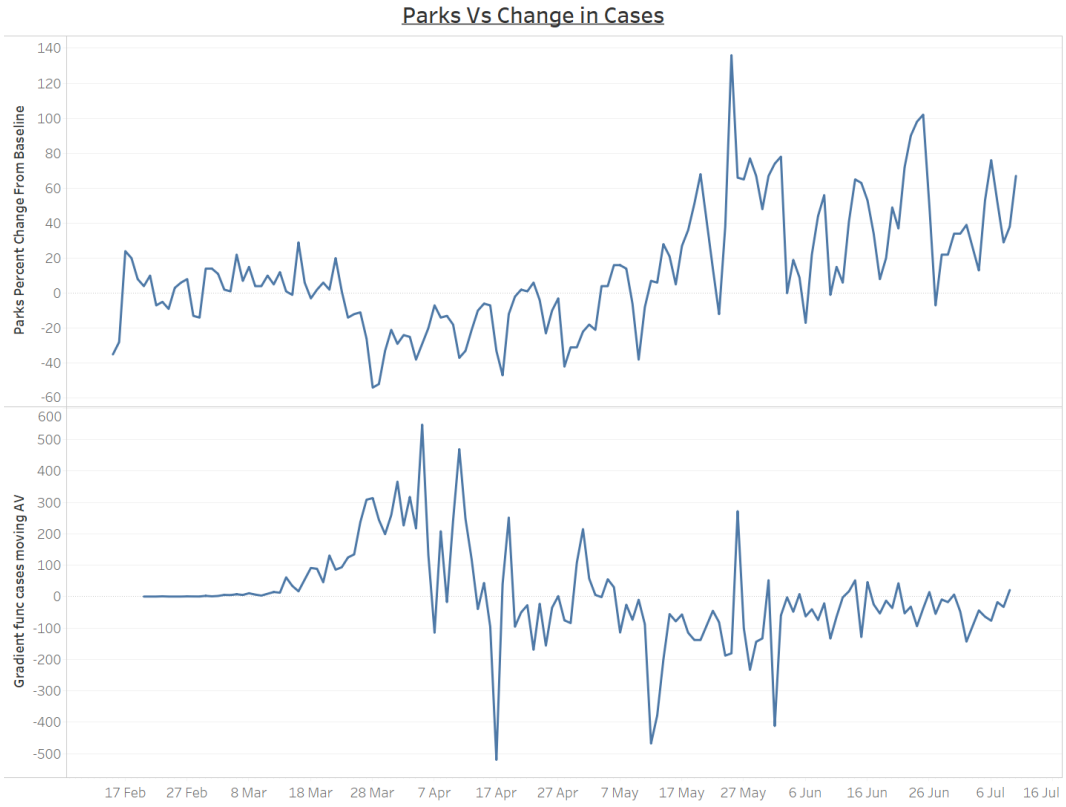
Grocery & Pharmacy Vs Change in Cases



Transit Vs Change in Cases




Residential Vs Change in Cases
Residential Percent Change From Baseline
30
25
20
15
10
5
0
Gradient func cases moving AV
600
500
400
300
200
100
0
-100
-200
-300
-400
-500
17 Feb
27 Feb
8 Mar
18 Mar
28 Mar
7 Apr
17 Apr
27 Apr
7 May
17 May
27 May
6 Jun
16 Jun
26 Jun
6 Jul
16 Jul


Parks Vs Change in Cases
Parks Percent Change From Baseline
140
120
100
80
60
40
20
0
-20
-40
-60
Gradient func cases moving AV
600
500
400
300
200
100
0
-100
-200
-300
-400
-500
17 Feb
27 Feb
8 Mar
18 Mar
28 Mar
7 Apr
17 Apr
27 Apr
7 May
17 May
27 May
6 Jun
16 Jun
26 Jun
6 Jul
16 Jul

From the above time series graphs, we can see that retail and recreation takes a sharp drop mid-march which was likely a result of the UK wide lockdown. The smoothed change in daily cases data seems to follow with a sharp drop 20-30 days later. The drop in smoothed daily cases also follows an initial increase from February to April, which accentuates the turnaround and suggests that the lockdown is proving effective. As the lockdown opens and retail begins to increase back towards normal, the drop in cases seems to stabilize also. This is also the case with other indicis such as grocery and pharmacy which is understandable as it is a similar variable to retail and recreation.

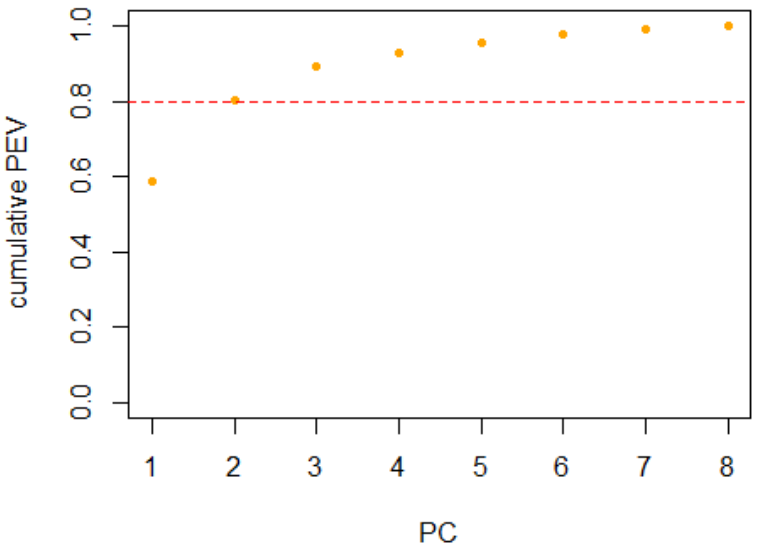
For residential percentage change, an increase in staying at home from the March UK lockdown seems to correlate with a decrease in cases which is consistent with the aforementioned lockdown theory. As the lockdown eases and people begin to leave their residences more, the decrease in cases stabilizes as it does with the two proceeding variables. It is a familiar story with the transit variable as well.

As for parks, people seem to visit them more often but there does not seem to be much correlation with new cases.

Principal Component Analysis:

Principal component analysis is a method of analysis which involves finding linear combinations of a set of variables that explain large variances in the data space. By finding the combinations that explain the largest variance, then the next largest and so on, these new combined variables have the potential to give a lot more insight into potential trends between data.

In order to create the principal components, the data was first centred and scaled, as required by the method. Upon creating the principal components, the standard values were used to calculate the proportion of explained variance. The cumulative proportion of explained variance was then plotted against the number of principal components to judge the value of each principal component. This plot can be viewed below:



As can be seen from the plot, the first two principal components explain 80% of the variance in the data, after which additional PC's add little value to the analysis. The first two principal components therefore have the highest potential of exposing a relationship.

The weightings of the principal components can be viewed below:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
new_cases	0.039733722	-0.705478069	0.046388	-0.02753	0.705043	-0.02563	0.002318	0.008001
new_deaths	0.066615461	-0.70015304	0.01705	-0.04126	-0.70476	0.071146	0.037643	0.01358
retail_and_recreation_percent_change_from_baseline	-0.440854797	-0.007562655	0.02096	0.194245	0.027969	0.371076	0.423305	0.670636
grocery_and_pharmacy_percent_change_from_baseline	-0.409404629	-0.089800356	-0.01408	0.771122	-0.04875	-0.42192	-0.15075	-0.16241
parks_percent_change_from_baseline	-0.307274341	-0.051073545	-0.87172	-0.23203	0.012783	-0.08848	0.210293	-0.19243
transit_stations_percent_change_from_baseline	-0.43417533	-0.020865338	0.115489	-0.01198	0.026989	0.690385	-0.31518	-0.46997
workplaces_percent_change_from_baseline	-0.402393493	0.00727576	0.468251	-0.34834	-0.02409	-0.32153	0.530773	-0.33429
residential_percent_change_from_baseline	0.432634083	0.029467134	-0.06656	0.435732	0.039894	0.301209	0.609413	-0.39238

From PC2 we can see that the variables contributing most to the variance of the principal component is new cases and new deaths (*the new deaths variable was still part of the analysis before I decided to delete it in the final dataset*). This is to be expected as we already know that the increase for these variables is exponential. As for PC1, we can see that there is consistency in loadings for retail and recreation, grocery and pharmacy, transit stations, workplaces and residential, with all of these variables showing a strong loading of +-0.4, which shows that they are in combination contributing equally to PC1. Thus as an interpretation of these loadings, we can see that as shopping goes down and as people travel less to work and stay more at home (as residential has a positive loading), then new_cases and deaths drop (derived from minus figures for these variables in PC2).

This result is a testimony that there is a relationship between human mobility and the spread of Covid-19, which was suggested in the initial review of the literature. To see if this relationship is significant, a linear regression was carried out on the data to yield its t values:

Variable significance:

In order to learn which of our variables are significant in the analysis and worth keeping, we need to employ a linear regression. There are some issues however which need addressing:

Multicollinearity:

Multicollinearity is the occurrence of high intercorrelations among independent variables in a multiple regression model. In general, multicollinearity can lead to wider confidence intervals and less reliable probability values for the independent variables {Kenton, Will 2020}. As can be seen by the below correlation matrix, the Android Mobility Indicis are highly correlated.

	New_cases	Retail & Rec	Grocery & Pharm	Parks	Transit	Work	Residential
New_cases	1.00	-0.07	0.02	-0.02	-0.05	-0.07	0.04
Retail & Rec	-0.07	1.00	0.85	0.60	0.90	0.80	-0.85
Grocery & Pharm	0.02	0.85	1.00	0.56	0.80	0.71	-0.76
Parks	-0.02	0.60	0.56	1.00	0.55	0.34	-0.60
Transit	-0.05	0.90	0.80	0.55	1.00	0.82	-0.86
Work	-0.07	0.80	0.71	0.34	0.82	1.00	-0.85
Residential	0.04	-0.85	-0.76	-0.60	-0.86	-0.85	1.00

All is not lost when it comes to linear regression as we can still use the method to help tell us which of our experimental variables (non-index variables) are significant in predicting new_cases.

Multicollinearity affects only the specific independent variables that are correlated. Therefore, if multicollinearity is not present for the independent variables that we are particularly interested in, we may not need to resolve it. Suppose our model contains the experimental variables of interest and some control variables. If high multicollinearity exists for the control variables but not the experimental variables, then we can interpret the experimental variables without problems. In our case, we know that the mobility indicis are highly correlated, as can be seen in the previous correlation matrix, but there were other non-correlated variables in the final dataset which we can test for significance, therefore it is still worth running an initial regression model.

Before we run our model, we will need to test for and correct potential heteroskedasticity. Heteroskedasticity is the phenomenon of non-constant variance of sampling errors. If the data is heteroskedastic, then it means the linear estimator is no longer the most efficient and there are other estimators that are better. i.e. weighted least squares or generalized least squares. I will therefore first test for heteroskedasticity using the Breusch Pagan test. If its positive, then I will need to use an alternative linear estimator, weighted least squares.

R Studio Breusch-Pagan test:

data: reg_all_varaibles_minus_region

BP = 129990, df = 13, p-value < 2.2e-16

As the P value is significantly small, it shows that there is heteroscedasticity present so I will need to correct for it in the test statistics. This was done using the coeftest function with white standard errors (Rstudio). The results can be viewed on the next page:

```
t test of coefficients:

                                                  Estimate  Std. Error  t value
(Intercept)                                     8.8683e+02  1.4781e+02   5.9997
total_cases                                     1.7028e-02  5.2124e-04  32.6677
stringency_index                               -5.4313e-01  4.7800e-01  -1.1363
population                                      2.5273e-06  2.0868e-07  12.1110
population_density                             -7.1717e-02  9.4078e-03  -7.6232
median_age                                      6.5444e+00  1.6273e+00   4.0217
gdp_per_capita                                  2.4684e-03  1.0577e-03   2.3337
life_expectancy                                -1.8116e+01  2.5232e+00  -7.1796
retail_and_recreation_percent_change_from_baseline -1.4072e+01  1.7912e+00  -7.8561
grocery_and_pharmacy_percent_change_from_baseline  1.4462e+01  1.2880e+00  11.2282
parks_percent_change_from_baseline             -5.0599e+00  3.6476e-01 -13.8718
transit_stations_percent_change_from_baseline   3.9047e-01  8.7283e-01   0.4474
workplaces_percent_change_from_baseline        -1.3231e+00  1.3828e+00  -0.9569
residential_percent_change_from_baseline       -3.4527e+00  2.9919e+00  -1.1540
                                                Pr(>|t|)
(Intercept)                                    2.021e-09 ***
total_cases                                    < 2.2e-16 ***
stringency_index                                 0.25586
population                                     < 2.2e-16 ***
population_density                             2.619e-14 ***
median_age                                     5.805e-05 ***
gdp_per_capita                                   0.01963 *
life_expectancy                                7.312e-13 ***
retail_and_recreation_percent_change_from_baseline 4.223e-15 ***
grocery_and_pharmacy_percent_change_from_baseline  < 2.2e-16 ***
parks_percent_change_from_baseline             < 2.2e-16 ***
transit_stations_percent_change_from_baseline    0.65462
workplaces_percent_change_from_baseline          0.33865
residential_percent_change_from_baseline         0.24851
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We can see that after correcting for heteroskedasticity in the model, it comes to light that all experimental variables (ignoring mobility indices) may be correlated with significant p values. The stringency index does not seem to be correlated which is odd, as in theory a higher stringency within a country should yield less cases. Given it reads insignificant however, it was deleted from the data.

Now that we have observed a relationship between our independent variables and new_cases, we are ready to model this relationship for future use cases.

Predictive Model

Before I proceeded to create the models, I had to address two inherent characteristics within the data which were volatility and time lag in the dependent variable. The volatility was addressed by using a moving average of 5 days. This helped get rid of the unexplained portion of the variance that was outside of the predictive scope of our independent variables. For our use purposes as well, it is more appropriate to use a moving average to avoid potential over fitting when training the model. For the

response lag issue, the machine learning models were fed three different datasets where the dependent variable was shifted back by 10, 20 and 30 days. This was inspired by the lag interpreted visually in the previously seen time series graphs.

Three Machine Learning algorithms were employed as models in my analysis. All models were created using the Jupyter integrated development environment and Python as the coding script.

Random Forest Application:

A random forest model was employed in my analysis as the first of three methods. Before training the model on the data I used, the region column had to be converted into a form understood by the model. The transformation was made using the "One Hot Encoder" class from the scikit learn python library. This class takes each category of data and gives it its own column with labels 0 and 1.

Once the data was ready for the model, it was split into the training and test set with the test set representing 20% of the dataset. The model was trained on the training set using 500 trees.

```
# Encoding the Region Column
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = (ct.fit_transform(X))

# Creating the Training/Test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

# Creating the Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 500, random_state = 0)
regressor.fit(X_train, y_train)
```

Support Vector Regression (SVR) Application:

A support vector regression was employed as the second of three machine learning methods to predict new_cases. The data was first normalized, as required by this method. After scaling the data, the region column was encoded as with the random forest model above. As with the random forest model, the model was trained using 80% of the dataset and tested on the remaining 20%. The RBF function was the kernel employed in the model:

```
# Scaling the Data
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X.iloc[:,1:] = sc_X.fit_transform(X.iloc[:,1:])
y = y.reshape(len(y),1)
sc_y = StandardScaler()
y = sc_y.fit_transform(y)

# Encoding the Region Column
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = (ct.fit_transform(X))

# Creating the Splits
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

# Training the Model
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X_train, y_train)
```

Artificial Neural Network (ANN) Application:

For the purposes of this essay and predicting daily cases of Covid-19, an artificial neural network was employed as the last of three methods in addition to the random forest and SVR.

The analysis was completed on the Google Collaboratory integrated development environment as I required the analysis to be completed on the cloud to allow me access to 16 gigabytes of ram with boosts of up to 32 gigabytes where required. An added advantage of Google Collaboratory is that many modules including the latest Tensorflow (the required module for ANN) already comes preinstalled.

I began by loading the required modules after which I uploaded the required datasets and cleared them of null values. I then needed to encode the regional data, of which label encoder was used:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
le.fit(dataset.region)
dataset['region'] = le.transform(dataset.region)
dataset
```

Artificial Neural Networks require independent variables to be standardized/normalized and so I proceeded to normalize the data and created a new data frame with the encoded regional data and dependent variable:

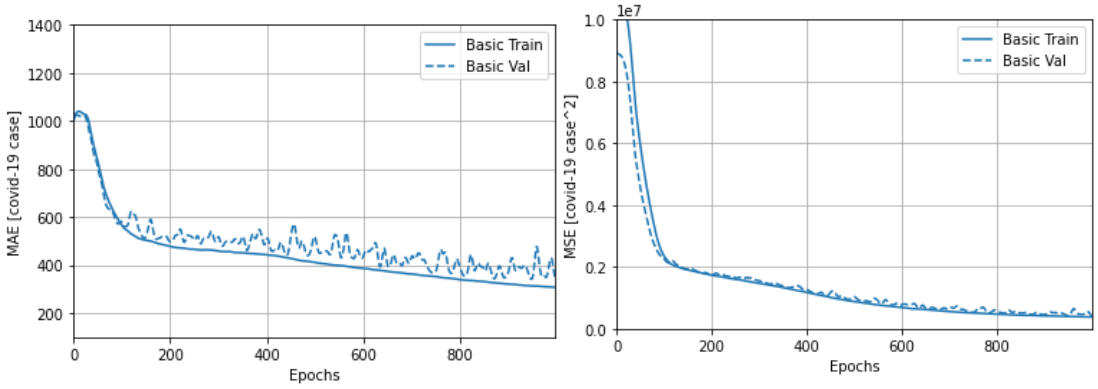
```
cln = [c for c in dataset.columns if c not in ['region', 'NewCase Moving Av 10 day FUT']]
# normalise variables without new cases and region variables
x = dataset.loc[:, cln]
x_norm = (x - x.mean()) / (x.max() - x.min())

# create a new DF and new cases and region
df = pd.DataFrame(x_norm)
df['NC'] = dataset['NewCase Moving Av 10 day FUT']
df.insert(0, 'region', dataset['region'])
```

I created the ANN using two hidden layers and using the rectifier (“relu”) activation function to help with computational efficiency and to help with issues of overfitting. Each hidden layer had 64 neurons. No activation function was used with the output node, as it was not required by the regression model. The optimizer used in the analysis was RMSprop which utilizes gradient descent but has the added advantage of reducing the steps required to converge towards the optimal values:

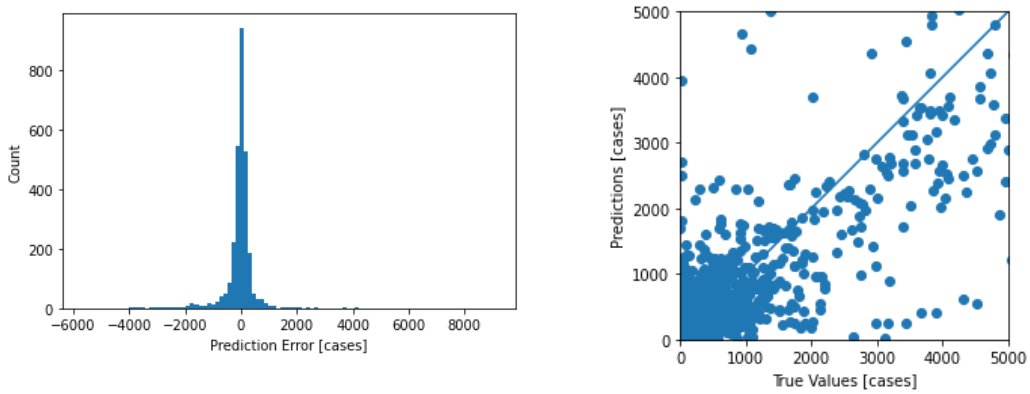
```
# designing the network

model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(64, activation='relu'))
```

```
model.add(Dense(1))

optimizer = RMSprop(0.001)

model.compile(loss='mse',
        optimizer=optimizer,
        metrics=['mae', 'mse']) # Mean Absolute Error (mae), Mean Squared Error (mse)
```

20% of the data was used for testing and the mean absolute error was used as the metric to evaluate the performance of the model. 1000 Epochs were used in the analysis to help the algorithm converge to the minimum loss level. By plotting the mean absolute error and mean squared error against the number of epochs we can see the learning process of the algorithm taking place. After 1000 epochs I did not see any fruitful decrease in error and so left the number of epochs constant at 1000.



The model was then evaluated on the test set using the predict method. The predicted values were plotted against the true values for visualisation purposes and the errors were also found to be normally distributed confirming the model is unbiased:



Results

To compare the results of the three different machine learning algorithms a measure was needed to compare the accuracy of the models.

R Squared:

The R Squared value was taken as the comparison value. R Squared is a measure that outputs the explained variance from the average Y value in the test set. It therefore can tell us how well our models are doing in explaining the changes in Y for changes in X (independent) variables:

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

Adjusted R Squared:

R squared suffers from an inherent problem that results in a higher R Squared value for increases in the number of independent variables. An extension of R Squared is therefore used when dealing with multiple input values. The formula can be viewed below:

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

Random Forest Results:

Lag (Days)	Adjusted R Squared Y test
10	0.994213791
20	0.992906252
30	0.984806018

10 Day			20 Day			30 Day		
Index	y_pred	y_test	Index	y_pred	y_test	Index	y_pred	y_test
0	697	597	0	372	367	0	370	348
1	444	613	1	356	374	1	6	3
2	18	21	2	3	0	2	0	0
3	179	181	3	278	297	3	8	7

4	31	40	**4**	269	330	**4**	13	14
...	...	...	**...**	...	...	**...**	...	...
2906	6	7	**2656**	47	46	**2456**	15	14
2907	274	302	**2657**	8	10	**2457**	28	26
2908	1	1	**2658**	318	301	**2458**	259	235
2909	770	768	**2659**	3805	3816	**2459**	44	30
2910	28	41	**2660**	2877	2923	**2460**	4735	5048

Support Vector Regression Results:

Lag (Days)	Adjusted R Squared Y test
10	0.962232026
20	0.943218483
30	0.897044637

10 Day			20 Day			30 Day		
Index	y_pred	y_test	Index	y_pred	y_test	Index	y_pred	y_test
0	-0.257747	-0.235145	**0**	-0.102263	-0.105328	**0**	0.618596	0.123892
1	-0.037551	-0.069982	**1**	-0.129229	-0.167571	**1**	-0.197921	-0.170789
2	-0.191691	-0.235659	**2**	-0.185759	-0.244537	**2**	-0.073718	-0.134476
3	-0.292212	-0.236121	**3**	-0.244521	-0.246605	**3**	-0.237174	-0.186501
4	-0.272338	-0.236789	**4**	-0.222714	-0.208639	**4**	1.169616	1.286903
...	...	...	**...**	...	...	**...**	...	...
2906	-0.077	-0.144935	**2656**	5.70768	7.238449	**2456**	-0.326285	-0.255614
2907	-0.119448	-0.224049	**2657**	-0.117617	-0.202287	**2457**	-0.278628	-0.209428
2908	-0.141368	-0.226001	**2658**	-0.173136	-0.234393	**2458**	-0.226726	-0.254807
2909	-0.186625	-0.232114	**2659**	-0.167877	-0.242961	**2459**	-0.282326	-0.255757
2910	-0.235508	-0.231446	**2660**	-0.134425	-0.226859	**2460**	-0.271469	-0.225852

Results Continue On Next Page

Artificial Neural Network Results:

Lag (Days)	Adjusted R Squared Y test
10	0.960258994
20	0.905150575
30	0.868945664

10 Day			20 Day			30 Day		
Index	y_pred	y_test	Index	y_pred	y_test	Index	y_pred	y_test
0	-72	57	0	1280	985	0	185	14
1	-130	0	1	-33	10	1	53	459
2	197	155	2	219	16	2	1297	559
3	505	269	3	1454	1942	3	283	86
4	212	567	4	3058	3401	4	-149	6
...	...	...	...	...	...	...	...	...
2906	21	402	2656	-57	47	2456	-6	187
2907	-2	79	2657	-326	11	2457	657	294
2908	799	612	2658	-151	435	2458	557	5
2909	651	1500	2659	21410	23014	2459	162	639
2910	127	84	2660	422	86	2460	102	301

Interpretation of Results:

Random Forest:

It can be seen from the R Squared figures that the trained model seems to explain well the variance of the errors with the model explaining over 98% of the variance at all levels of lag. The model with a 10-day lag seemed to fit best and perform best with the highest figure of 99.4%.

If we look at the sample of predicted vs actual values as well, the numbers seem to match near perfectly on all occasions.

The random forest therefore seems to do a great job in modelling the change in cases for Covid-19.

Support Vector Regression (SVR):

The SVR did not perform as well with adjusted R Squared values ranging from 89% to 96%. It did follow the same pattern as the Random Forest in that the 10 day lag dataset performed best, then the 20 and 30 respectively.

Artificial Neural Network (ANN):

The ANN performed the least strongly with R squared values for 10, 20 and 30 days at 86%, 90% and 96% respectively. It followed the same pattern as the first two methods with the 10 day dataset performing the best. This method however did output negative values as can be seen in its sample output. In hindsight I would have explored activation functions that avoid a negative output.

Concluding Discussion

Discussion of Findings & Aims:

As mentioned at the beginning of this report, the aims of my project were two-fold. The first was to model the spread of the disease, to allow us to intervene earlier in future scenarios. The second aim of the project was to see whether regional lockdowns prove an effective means to curb the disease, and this aim would be fulfilled if we could create an effective model by using Google's Global Mobility Report.

As this research project was successful in its aim to model the change in cases of Covid-19, (suggested by the high R Squared figures given in the results section), I can conclude that my aims for the project on a whole were largely met. The success of model creation suggests that there is a strong relationship between mobility and spread rate of Covid-19, and this report lends evidence to the belief that lockdown policy does curb the spread of Covid-19.

The fact that the 10-day lag model performed best can also be interpreted as being in line with current literature around Covid-19. This is because, as mentioned in the review of the literature, the incubation period mentioned was approximately 5.2 days {Rothan, Hussin A 2020}. Therefore, changes in mobility should see confirmed cases begin to fall before 10 days. With this in mind, It would be interesting to see how a weekly, bi-weekly and tri-weekly model may have performed.

Although I am pleased with the results, I do believe this report could have been carried out in a mode much more informative for a variety of reasons.

Regional Analysis:

My original intention was to create a UK based model to model the spread around the UK. I believe this would have been more useful as it could have informed more intricate policy decisions around the pandemic. The original country level cases dataset had subregional data for the UK and so did the Android Global Mobility Index, however the cases in both these datasets for subregional UK data was grouped differently. i.e. one dataset grouped cases by council and others by borough and I therefore could not merge on these fields, so had to stick with a global model. This made the dataset a lot noisier and difficult to interpret due to the multi-regional, cultural, and political variables that are present in different countries.

Multi-regional analysis also meant there were multiple dimensions to the analysis and made it difficult to carry out proper Exploratory analysis. Exploratory analysis relies heavily on visual interpretation and I had to focus on UK data to find an informative trend in the data. Plotting the whole dataset was too noisy an analysis and would not provide any informative output.

Improvements to ML Models:

I found myself rerunning the analysis several times when revising the model and getting different results. In hindsight I could have used the same random seed across the analysis for both training and testing so that all three models would have been trained and tested on the same data.

This would have been especially useful when tuning my ANN as results can vary widely for the same model depending on where the seed is set. This made it hard to know if my model was better or worse after tuning.

In hindsight I realized that I could have used the transformed values from the first two principal components within the machine learning model. The first principal component was a reflection of the whole Android Mobility Index, as several of the indicis had significant weights in the loadings. This may have ridded the model of some of the noise element and prevented issues associated with multicollinearity which means I could have utilized a linear regression as a fourth method.

Recommendations for Future Research:

As touched upon above, regional analysis may be more useful than global analysis and this can be made possible through conformity of the grouping methodology used by data collectors.

On a global level, I believe it will be very useful to factor in major transport hubs into the analysis such as airports as they are necessary for travel and movement. The UK civil aviation authority for example has publicly available data on all public flights that enter and leave all UK airports. This data however is grouped and published as monthly data. Daily data of this type would have been largely informative as it could be listed against changes in daily cases of Covid-19 and modelled. This would give us useful information on how the disease is travelling.

Personal Reflections:

The completion of this dissertation project has allowed me to heavily develop my knowledge around machine learning and its application in the real world. My coding ability grew massively, and I became familiar with a range of software such as Python, RStudio and Tableau. I also learned much about cloud computing technology having been introduced to platforms such as Google Collaboratory and Databiology. I had to account for time in training models and had to consider processing power when creating my model. This project has opened me up to exploring Data Science more rigorously, as I know I have only just scratched the surface of what it can do.

References

Available at: https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1? (Accessed: 09/13 2020).

Adhikari, S.P., Meng, S., Wu, Y., Mao, Y., Ye, R., Wang, Q., Sun, C., Sylvia, S., Rozelle, S. and Raat, H. (2020) 'Epidemiology, causes, clinical manifestation and diagnosis, prevention and control of coronavirus disease (COVID-19) during the early outbreak period: a scoping review', *Infectious diseases of poverty,* 9(1), pp. 1-12.

Gan, G., Ma, C. and Wu, J. (2007) *Data clustering: theory, algorithms, and applications.* SIAM.

Hu, Y., Sun, J., Dai, Z., Deng, H., Li, X., Huang, Q., Wu, Y., Sun, L. and Xu, Y. (2020) 'Prevalence and severity of corona virus disease 2019 (COVID-19): A systematic review and meta-analysis', *Journal of Clinical Virology,* , pp. 104371.

JIANG, N. and RYAN, J. (2020). Available at: https://blogs.worldbank.org/developmenttalk/how-does-digital-technology-help-fight-against-covid-19 (Accessed: 09/13 2020).

Kenton, W. (2020) *Fundamental Analysis: Multicollinearity*. Available at: https://www.investopedia.com/terms/m/multicollinearity.asp (Accessed: 09/20 2020).

Khachfe, H.H., Chahrour, M., Sammouri, J., Salhab, H., Makki, B.E. and Fares, M. (2020) 'An epidemiological study on COVID-19: a rapidly spreading disease', *Cureus,* 12(3).

Lake, M.A. (2020) 'What we know so far: COVID-19 current clinical knowledge and research', *Clinical medicine (London, England),* 20(2), pp. 124-127.

Lipsitch, M., Swerdlow, D.L. and Finelli, L. (2020) 'Defining the epidemiology of Covid-19â€"studies needed', *New England journal of medicine,* 382(13), pp. 1194-1196.

Park, M., Cook, A.R., Lim, J.T., Sun, Y. and Dickens, B.L. (2020) 'A systematic review of COVID-19 epidemiology based on current evidence', *Journal of Clinical Medicine,* 9(4), pp. 967.

Peeri, N.C., Shrestha, N., Rahman, M.S., Zaki, R., Tan, Z., Bibi, S., Baghbanzadeh, M., Aghamohammadi, N., Zhang, W. and Haque, U. (2020) 'The SARS, MERS and novel coronavirus (COVID-19) epidemics, the newest and biggest global health threats: what lessons have we learned?', *International journal of epidemiology,* .

Rothan, H.A. and Byrareddy, S.N. (2020) 'The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak', *Journal of Autoimmunity,* , pp. 102433.

Surveillances, V. (2020) 'The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (COVID-19)â€"China, 2020', *China CDC Weekly,* 2(8), pp. 113-122.

Various (2020). Available at: https://en.wikipedia.org/wiki/Neuron (Accessed: 09/25 2020).

Walsh, F. (2020). Available at: https://www.bbc.co.uk/news/world-europe-53735718 (Accessed: 09/13 2020).

Wang, Y., Wang, Y., Chen, Y. and Qin, Q. (2020) 'Unique epidemiological and clinical features of the emerging 2019 novel coronavirus pneumonia (COVIDâ€19) implicate special control measures', *Journal of medical virology,* 92(6), pp. 568-576.

WHO (2020). Available at: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it#:~:text=The%20International%20Committee%20on%20Taxonomy,two%20viruses%20are%20different. (Accessed: 09/13 2020).

World Bank (2020). Available at: https://www.worldbank.org/en/news/feature/2020/06/08/the-global-economic-outlook-during-the-covid-19-pandemic-a-changed-world (Accessed: 09/13 2020).

Appendix

Data Merging Code:

```
                                    # Data Merging & Cleaning

import numpy as np
import pandas as pd
import os as os
import matplotlib.pyplot as plt
# Read in Datasets
cases = pd.read_csv("Country Level Cases Dataset.csv")
os.chdir("C:/Users/nassi/OneDrive/Data Science Masters/2018 Course/Disso/Covid 19 Datasets/Up to Da
mobility = pd.read_csv("Android Global_Mobility_Report.csv")
# Merge Datasets
merged_dataset = pd.merge(left = cases,right = mobility,how = "inner", on = ["region","date"])
# Output Null Values
null_values = merged_dataset1.isna()
null_values = null_values.sum()
null_values = null_values / len(merged_dataset1)
null_values = null_values * 100
# Correlation Matrix
correlation = merged_dataset.corr()
```

Exploratory Analysis Code (Next Page):

```r
# Upload Data
read.csv('')

# Histograms
hist(my_data$total_cases, main = "Total Cases")
hist(my_data$new_cases, main = "New Cases")
hist(my_data$total_deaths, main = "Total Deaths")
hist(my_data$new_deaths, main = "New Deaths")
hist(my_data$total_cases_per_million, main = "Total Cases Per Mil")
hist(my_data$new_cases_per_million, main = "New Cases Per Mil")
hist(my_data$total_deaths_per_million, main = "Total Deaths Per Mil")
hist(my_data$new_deaths_per_million, main = "New Deaths Per Mil")
hist(my_data$retail_and_recreation_percent_change_from_baseline, main = "Retail & Recreation")
hist(my_data$grocery_and_pharmacy_percent_change_from_baseline, main = "Grocery & Pharmacy")
hist(my_data$parks_percent_change_from_baseline, main = "Parks")
hist(my_data$transit_stations_percent_change_from_baseline, main = "Transit Stations")
hist(my_data$workplaces_percent_change_from_baseline, main = "Workplaces")
hist(my_data$residential_percent_change_from_baseline, main = "Residential")

# Boxplots
boxplot(my_data$total_cases, main = "Total Cases")
boxplot(my_data$new_cases, main = "New Cases")
boxplot(my_data$total_deaths, main = "Total Deaths")
boxplot(my_data$new_deaths, main = "New Deaths")
boxplot(my_data$total_cases_per_million, main = "Total Cases Per Mil")
boxplot(my_data$new_cases_per_million, main = "New Cases Per Mil")
boxplot(my_data$total_deaths_per_million, main = "Total Deaths Per Mil")
boxplot(my_data$new_deaths_per_million, main = "New Deaths Per Mil")
boxplot(my_data$retail_and_recreation_percent_change_from_baseline, main = "Retail & Recreation")
boxplot(my_data$grocery_and_pharmacy_percent_change_from_baseline, main = "Grocery & Pharmacy")
boxplot(my_data$parks_percent_change_from_baseline, main = "Parks")
boxplot(my_data$transit_stations_percent_change_from_baseline, main = "Transit Stations")
boxplot(my_data$workplaces_percent_change_from_baseline, main = "Workplaces")
boxplot(my_data$residential_percent_change_from_baseline, main = "Residential")
```

```r
# Scatter Graphs (completed on Tableau)

# Principal componant analysis:
  # first create new dataset with only cases, deaths and mobility incicis
data_4_princ_comp <- my_data[,c(6,8,16:21)]
  # create Principal Componants
    # note: variables are centered and scaled before analysis
pc_data <- prcomp(data_4_princ_comp, center = T, scale. = T)
  # calculate the proportion of explained variance (PEV) from the std values
pc_data_var <- pc_data$sdev^2
pc_data_var
pc_data_PEV <- pc_data_var / sum(pc_data_var)
pc_data_PEV
  # plot the cumulative value of PEV for increasing number of additional PCs
  #   note: add an 80% threshold line to inform the feature extraction
opar <- par()
plot(
  cumsum(pc_data_PEV),
  ylim = c(0,1),
  xlab = 'PC',
  ylab = 'cumulative PEV',
  pch = 20,
  col = 'orange'
)
abline(h = 0.8, col = 'red', lty = 'dashed')
par(opar)

  # get and inspect the loadings for each PC
pc_data_loadings <- pc_data$rotation
pc_data_loadings
write.csv(pc_data_loadings,'pc loadings2.csv')
```

```r
# will conduct linear regression analysis to check for siginificance of columns/factors/varaibles
# first need to create the model
lin_model <- lm(new_cases ~ total_cases + stringency_index + population + population_density + med
reg_all_varaibles_minus_region = summary(lin_model)
write.csv(reg_all_varaibles_minus_region, file = "regression minus region.csv")


# second need to test for heteroskedasticity and if so, correct for it.
bptest(reg_all_varaibles_minus_region, studentize = FALSE)
library(sandwich)
#linear coefficient model adjusted for heteroskedasticity
coeftest(lin_model, vcov = vcovHC(lin_model, "HC1"))
```

Random Forest Model Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os as os

# Importing The Dataset
os.chdir('C:/Users/nassi/OneDrive/Data Science Masters/2018 Course/Disso/Covid 19 Datasets/Python Sc
data_10 = pd.read_csv('Dataset 10 day.csv')
# Clearning the Dataset of Null Values
data_10.dropna(inplace = True)
# Encoding the Region Column
X = data_10.iloc[:, 0:-1].values
y = data_10.iloc[:, -1].values
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = (ct.fit_transform(X))
# Creating the Training & Testing Splits
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
# Creating & Training the Random Forest
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 500, random_state = 0)
regressor.fit(X_train, y_train)
# Predicting & Viewing New Results
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
Results_data = np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1)
CoL = ("y_pred","y_test")
pd.DataFrame(data = Results_data, columns = CoL)
# Computing R Squared Values
from sklearn.metrics import r2_score
r2 = r2_score(y_true = y_test, y_pred = y_pred)
Adj_r2 = 1-((1-r2)*(len(y_pred)-1)/(len(y_pred)-13-1))
```

Support Vector Regression Model Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os as os

# Importing The Dataset
os.chdir('C:/Users/nassi/OneDrive/Data Science Masters/2018 Course/Disso/Covid 19 Datasets/Python Scrips/
data_30 = pd.read_csv('Dataset 30 day.csv')
# Deleting Null Values
data_30.dropna(inplace = True)
# Scaling The Data
X = data_30.iloc[:, 0:-1]
y = data_30.iloc[:, -1].values
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X.iloc[:,1:] = sc_X.fit_transform(X.iloc[:,1:])
y = y.reshape(len(y),1)
sc_y = StandardScaler()
y = sc_y.fit_transform(y)
# Encoding the Region Column
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = (ct.fit_transform(X))
# Creating the Splits
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
# Training the Model
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X_train, y_train)
# Predicting Y Test
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
Results_data = np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1)
CoL = ("y_pred","y_test")
pd.DataFrame(data = Results_data, columns = CoL)
# Computing R Squared
from sklearn.metrics import r2_score
r2 = r2_score(y_true = y_test, y_pred = y_pred)
Adj_r2 = 1-((1-r2)*(len(y_pred)-1)/(len(y_pred)-135-1))
```

Artificial Neural Network Model Code:

```
import tensorflow as tf
print(tf.__version__)

# downloading tensorflow docs
!pip install -q git+https://github.com/tensorflow/docs

# importing modules
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import RMSprop, Adam, SGD
from sklearn.model_selection import train_test_split
from keras import regularizers

import tensorflow_docs as tfdocs
import tensorflow_docs.plots
import tensorflow_docs.modeling
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Importing Dataset
dataset = pd.read_csv('Dataset 10 day.csv')
# Deleting Null Values
dataset.dropna(inplace = True)
# Encoding the Region Column
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
le.fit(dataset.region)
dataset['region'] = le.transform(dataset.region)

cln = [c for c in dataset.columns if c not in ['region', 'NewCase Moving Av 10 day FUT']]
# normalising variables without new cases and region variables
x = dataset.loc[:, cln]
x_norm = (x - x.mean()) / (x.max() - x.min())

# creating a new DF and new cases and region
df = pd.DataFrame(x_norm)
df['NC'] = dataset['NewCase Moving Av 10 day FUT']
```

```
# creating a new DF and new cases and region
df = pd.DataFrame(x_norm)
df['NC'] = dataset['NewCase Moving Av 10 day FUT']
df.insert(0, 'region', dataset['region'])

X = df.iloc[:, :-1]
y = df.iloc[:, -1]

# splitting the data to training and test set
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=0)

# designing the network

model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(1))

optimizer = RMSprop(0.001)

model.compile(loss='mse',
              optimizer=optimizer,
              metrics=['mae', 'mse']) # Mean Absolute Error (mae), Mean Squared Error (mse)

model.summary()

# validation is taken from the training set to check the accuracy as the model is learning

EPOCHS = 1000

history = model.fit(
  X_train, y_train,
  epochs=EPOCHS, validation_split=0.2, verbose=0,
  callbacks=[tfdocs.modeling.EpochDots()])

# geting the info of each epoch for later use
hist = pd.DataFrame(history.history)
hist['epoch'] = history.epoch
hist.tail()
```

```
# geting the info of each epoch for later use
hist = pd.DataFrame(history.history)
hist['epoch'] = history.epoch
hist.tail()

# Plotting the loss using the mean absolute error
plotter = tfdocs.plots.HistoryPlotter(smoothing_std=3)
plotter.plot({'Basic': history}, metric="mae")
plt.ylim([100, 1400])
plt.ylabel('MAE [covid-19 case]')
# Plotting the loss using the mean standard error
plotter.plot({'Basic': history}, metric="mse")
plt.ylim([0, 10000000])
plt.ylabel('MSE [covid-19 case^2]')

# evaluting the model using the test set
loss, mae, mse = model.evaluate(X_test, y_test, verbose=2)
print("Testing set Mean Abs Error: {:5.2f} number of covid-19 cases".format(mae))
print("Testing set Mean Squared Error: {:5.2f} number of covid-19 cases".format(mse))

# making predictions using the data in the test set

test_predictions = model.predict(X_test).flatten()

a = plt.axes(aspect='equal')
plt.scatter(y_test, test_predictions)
plt.xlabel('True Values [cases]')
plt.ylabel('Predictions [cases]')
lims = [0, 5000]
plt.xlim(lims)
plt.ylim(lims)
_ = plt.plot(lims, lims)

# checking the error distribution

error = test_predictions - y_test
plt.hist(error, bins = 100)
plt.xlabel("Prediction Error [cases]")
_ = plt.ylabel("Count")

# calculating R Squared Values
```

```
plotter.plot({'Basic': history}, metric="mse")
plt.ylim([0, 10000000])
plt.ylabel('MSE [covid-19 case^2]')

# evaluting the model using the test set
loss, mae, mse = model.evaluate(X_test, y_test, verbose=2)
print("Testing set Mean Abs Error: {:5.2f} number of covid-19 cases".format(mae))
print("Testing set Mean Squared Error: {:5.2f} number of covid-19 cases".format(mse))

# making predictions using the data in the test set

test_predictions = model.predict(X_test).flatten()

a = plt.axes(aspect='equal')
plt.scatter(y_test, test_predictions)
plt.xlabel('True Values [cases]')
plt.ylabel('Predictions [cases]')
lims = [0, 5000]
plt.xlim(lims)
plt.ylim(lims)
_ = plt.plot(lims, lims)

# checking the error distribution

error = test_predictions - y_test
plt.hist(error, bins = 100)
plt.xlabel("Prediction Error [cases]")
_ = plt.ylabel("Count")

# calculating R Squared Values
from sklearn.metrics import r2_score
r2 = r2_score(y_test,test_predictions)
Adj_r2 = 1-((1-r2)*(len(y_test)-1)/(len(y_test)-13-1))
Adj_r2

# viewing y test vs y pred values
y_test = y_test.to_numpy()
np.set_printoptions(precision=2)
Results_data = np.concatenate((test_predictions.reshape(len(test_predictions),1), y_test.reshape(len(y_test),1)),1)
CoL = ("y_pred","y_test")
pd.DataFrame(data = Results_data, columns = CoL)
```