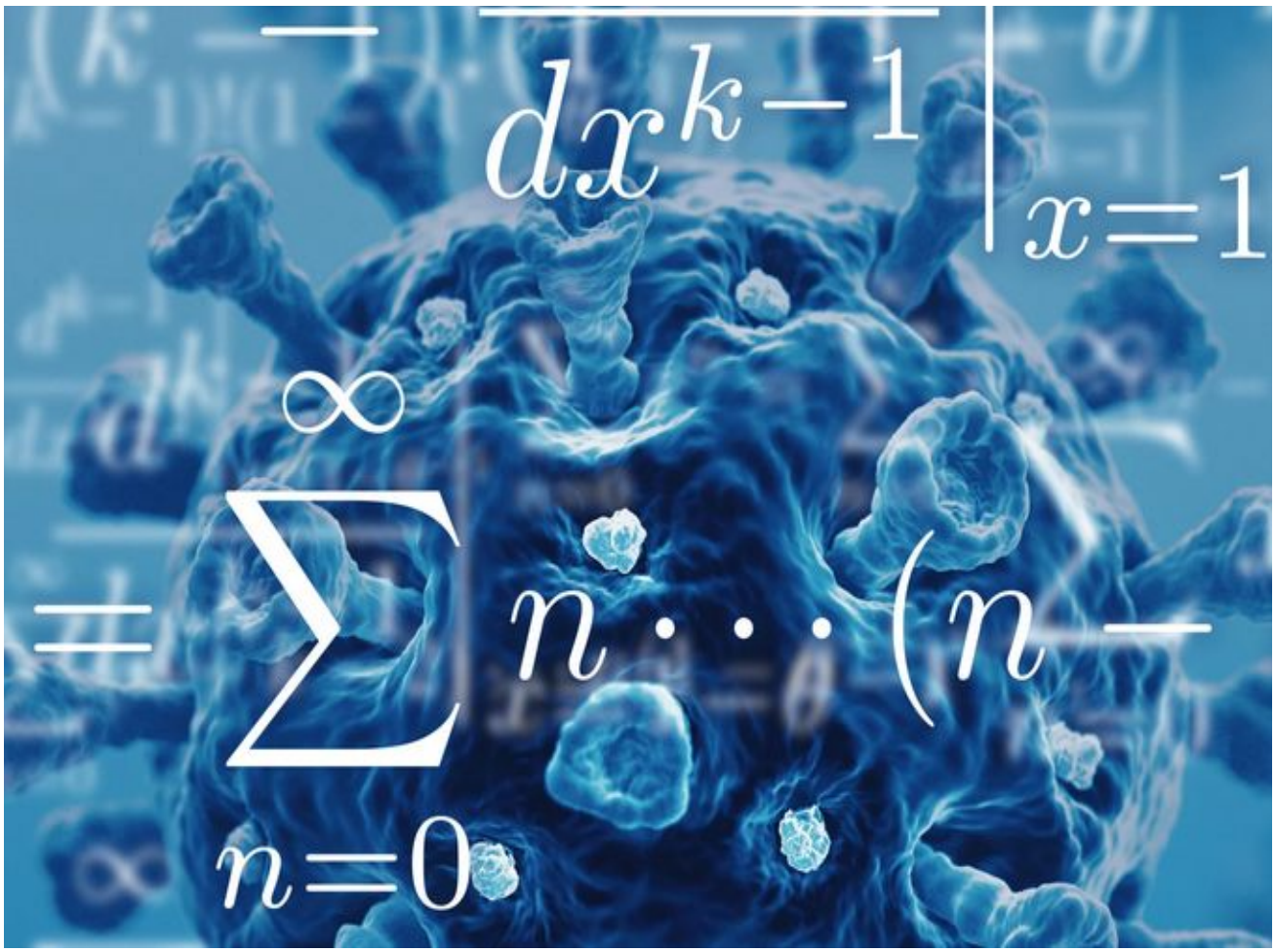
IBM Data Science Professional Certificate

Applied Data Science Capstone, July 2020

Capstone Project - The Battle of Neighborhoods

PREDICTION OF THE COVID-19 CONTAGION INTENSITY IN CHICAGO, IL.

Mutuyimana MANZI



1. INTRODUCTION

1.1. Background

On March 21, 2020, the City of Chicago came under the stay-at-home gubernatorial order due to the Coronavirus Disease in the last quarter of 2019 (COVID-19). As a disease that is menacing the health of the world population, there is a necessity to predict what could be its contagion intensity based on preexisting factors. Hence, it is crucial to take into consideration that health is about more than just physical well-being. Health is also determined by social and economic factors, political policies, the environment, behaviors, as well as health care quality and access.

Therefore, the fact that contagion intensity of COVID-19 is situation-dependent, it makes it not only being affected by the properties of the pathogen -such as how infectious it is-, but also by routines of the host population – for instance, how susceptible people are due to nutritional status or other illnesses that may compromise one's immune system. Moreover, the contagion intensity of COVID-19 may also be affected by the environment, demographics, socioeconomic factors and motives of administrative polities.

Elsewhere, the application of machine learning's algorithm of deep neural network to predict continuous values - such COVID-19 contagion intensity-, it oftentimes faces an insurmountable hurdle of vanishing gradient descent and/or local minima and maxima.

1.2. Problem

With the contagion intensity of COVID-19 serving as the target variable, this study aims to develop a supervised neural network model that predicts the intensity of

the contagion of the coronavirus in Chicago per zip code area on the basis of the preexisting factorial data such as data of clinic care, of health behavior, of mortality, of morbidity, of socioeconomic factors, of administrative policies, of physical environment, of demography, of public safety and of education. The COVID-19 specific properties such as positivity rate, test rate, death rate, transmission rate and coronavirus contamination pathways are used to calculate the actual contagion intensity.

This study is structured on three principal components, namely; dataset engineering, exploratory data analysis and development of the supervised neural network to predict continuous values of COVID-19 contagion intensity per Chicago's zip code areas.

In order to surmount the hurdle of local minima and maxima, and vanishing gradient descent oftentimes encountered while applying the machine's learning algorithms of deep neural networks to predict continuous values, this project puts in place a technique of a two-part hybrid deep neural network. That hybridation applies algorithms featuring the utilization of Swish and Softplus functions and of Rectified Linear Units (ReLU) and leaky Rectified Linear Units (leaky ReLU) functions. In both cases, those functions work as activation functions, whereby the latter pair are applied in the second part of the neural network as the former are in the first part of the neural network.

1.3. Interest

In the case of any looming coronavirus pandemic, the values of contagion intensity yielded by the predictive machine learning algorithms would help policy makers, healthcare officials, medical staff and administrative polities to put in place preventive and effective blueprint appropriate with every zip code area in Chicago, in order to curb the spread of coronavirus. At the other end of the balance, those

values would serve as an informative medium for the host population so that they can undertake protective steps.

At the other end of balance, the application of the hybrid of swish, softplus, ReLU and leaky ReLU as the activation functions in deep neural network to predict continuous values, it may appeal to the interest of machine learning developers facing the issue of vanishing gradient descent as well as local minima and maxima along the way.

2. DATA

The data used for this project emanates from a wide range of techniques and sources that include the City of Chicago Data Portal, different websites, open source databases and data garnering by the author. The data used for the target variable is obtained from the database of the City of Chicago on COVID-19 (City of Chicago, 2020). That dataset is made of 21 columns and 1000 rows.

The variable of reference to be used as the index in the dataset consists of the zip code areas in the City of Chicago. The zip code areas obtained from the website of the City of Chicago (City of Chicago, 2020) show that there are 59 zip codes. Some of the available data are organized per community area. Therefore, a community area column is embedded in the principal dataframe based on the dataset available on the website of Fulton Grace (Fulton Grace, 2020).

The principal pandas data frame is updated by an additional dataset made of all of Chicago's City zip codes as indexes and 30 columns. The dataset about the Chicago Police Department (CPD) Stations is used as a measure of public infrastructure and safety as well as government response. The principal data frame "datum" is updated with six columns from the CPD webpage on the Chicago Data Portal (City of Chicago, 2020).

The "datum" data frame is updated with the column "illegitimate_police_stops", which is about the percentage of adults who perceived no legitimate reason for their most recent stop by the police. Those data points obtained from the Sinai Community Health Survey (Sinai Community Health Survey, 2015-2016), It is an estimation percent of adults (aged ≥18 years) who reported that they had been stopped by the police in their lifetime and that the police did not have a legitimate reason for stopping them at their most recent stop. This percent is weighted to represent the population of adults aged ≥18 years who reside within the listed community area.

Table 2.1. *CPD's stations data*

Column Name	**Data Type**	**Measure Description**
|dist_close_cpd	float64	The shortest distance between a CPD station and the closest zip code area
cpd_district	int64	CPD District Number
cpd_district_name	str	CPD District name
cpd_zip	str	CPD zip code
cpd_longitude	float64	CPF longitude location
cpd_latitude	float64	CPD latitude location

The Chicago Fire Department (CFD) stations are used as a measure of public investment, infrastructure and safety. Data obtained from the Chicago Data Portal (City of Chicago, 2020) updates the "datum" data frame with four columns.

Table 2.2. *CFD stations data*

Column Name	Data Type	Measure Description
fire_name	str	Name of the CFD fire station
fire_engine	list	Names of the CFD fire engines
fire_num	int64	The number of fire engines in a given zip code area
fire_percent	float64	The number of fire engines per 100 residents

The hospital facilities serve as a showcase for health and human services as well as the infrastructure. The data is obtained through the means of using data available from the api of foursquare.com, and the names the author obtained and organized into the list. The "datum" data frame is updated with five columns.

Table 2.3. *Hospitals data*

Column Name	Data Type	Measure Description
hospital_num	int64	Number of hospitals in a zip code
hospitals_in_zip	list	Names of the hospitals
hospital_percent	float64	The number of hospitals per 100 residents in a zip code

Data about clinics in Chicago is a dataset obtained from City of Chicago mental health, sexually transmitted infection (STI) specialty, and women infant children

(WIC) clinic locations, hours of operation and contact information (City of Chicago, 2020).

Clinics in Chicago are included in this project as an indicator of infrastructure investment, health and human services. Ultimately, the "datum"] data frame is updated with "clinic_num", "clinics_in_zip", and "clinics_percent" columns, which are respectively the integer number of clinics in a given zip code, the list of sites in a given zip code, and the float number of sites per one hundred residents.

The dataset of primary care facilities is made of locations and contact information for Chicago primary care community health clinics (including all federally qualified health centers and similar community health centers that provide primary care and are open to the general community). The data is obtained from the data portal of the City of Chicago (City of Chicago, 2020). Primary care facilities are included in this project as an indicator of infrastructure investment, health and human services. Therefore, the "datum" data frame is updated with "pcare_num", "pcare_in_zip", and "pcare_percent" columns, which are respectively the integer number of primary care clinics in a given zip code, the list of primary care in a given zip code, and the float number of primary care facilities per one hundred residents.

Beside being an indicator of infrastructure investment, health and human services, nursing homes are of particular interest given the fact that those facilities are still operational during the COVID-19 lockdown. And residents of nursing homes live in close proximity, while a considerable number of staff members come in and go out on a somehow regular basis. The data obtained from the website of Dibbern(Dibbern, 2020) makes the "datum" data frame is thus updated with "nh_num", "nh_in_zip", and "nh_percent" columns, which are respectively the integer number of nursing homes in a given zip code, the list of nursing homes in a given zip code, and the float number of nursing homes per one hundred residents.

The testing sites of COVID-19 figures among the indicators of public policy response and health infrastructure. From the dataset containing specifics about the COVID-19 testing sites (City of Chicago, 2020), two columns -"address" and "facility" are used to update the "datum" principal data frame. The update is done with three columns as it is shown in Table 2.4.

Data about Chicago Public Schools (CPS) is included as an indicator of infrastructure and education service. The dataset comprises the locations of educational units in the Chicago Public School District for the school year 2019-2020 (City of Chicago, 2020). With two columns -"sch_addr" and "school_nm" are used to update the "datum" principal dataframe, whereby the new columns of "datum" principal data frame are "cps_num", "cps_in_zip", and "cps_percent".

Table 2.4. *COVID-19 testing sites data*

Column Name	Data Type	Measure Description
covid_num	int64	Number of COVID-19 testing sites in a zip code
covid_test_in_zip	list	Names of COVID-19 testing sites in a zip code
covid_test_percent	float64	The number of COVID-19 testing sites per 100 residents in a zip code

Data about Chicago Public Schools (CPS) is included as an indicator of infrastructure and education service. The dataset comprises the locations of educational units in the Chicago Public School District for the school year 2019-2020 (City of Chicago, 2020). With two columns -"sch_addr" and "school_nm" are used to update the

"datum" principal dataframe, whereby the new columns of "datum" principal data frame are "cps_num", "cps_in_zip", and "cps_percent".

Data about traffic crashes is obtained from the City of Chicago (City of Chicago, 2020). This data contains information about people involved in a crash and if any injuries were sustained.

The Problem Landlord List Buildings (City of Chicago, 2020) contains data in the form of a list describing landlords and property owners who are designated "problem landlords". Landlords on this list have had two or more administrative hearing causes brought against them and were found liable or defaulted to one or more serious building violations.

Crimes - 2001 to Present (City of Chicago, 2020) is the dataset reflecting reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago from 2001 to present, minus the most recent seven days. Data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system.

Public Libraries (City of Chicago, 2020) is a dataset of public libraries in Chicago. Libraries make up part of public investment, education and infrastructure. The libraries' dataset contains locations, hours and contact information.

Business Licenses - Current Active (City of Chicago, 2020) is the dataset containing all current and active business licenses issued by the Chicago's Department of Business Affairs and Consumer Protection.

Affordable Rental Housing Developments (City of Chicago, 2020) is a dataset in the category of Community and Economic Development. This dataset contains the list of affordable units/rental housing developments that are supported by City of Chicago programs to maintain affordability in local neighborhoods.

Grocery Stores - 2013 (City of Chicago, 2020), is the dataset containing a list of grocery stores which was used by the city to calculate the estimates of Chicagoans living in food deserts in 2013.

Micro-Market Recovery Program (MMRP) (City of Chicago, 2020) dataset contains all Department of Buildings (DOB) Permits that have occured at properties falling within any MMRP Zone. The City of Chicago launched the Micro-Market Recovery Program (MMRP), a coordinated effort among the City, not-for-profit intermediaries, and nonprofit and for-profit capital sources to improve conditions, strengthen property values, and create environments supportive of private investment in targeted markets throughout the city. The goal of MMRP is to improve conditions, strengthen property values, and create environments supportive of private investment in targeted areas by strategically deploying public and private capital and other tools and resources in well-defined micro-markets.

Tax Increment Financing (TIF) Funded RDA and IGA Projects (City of Chicago, 2020) is a dataset containing information on TIF-funded Redevelopment Agreements (RDA) and Intergovernmental Agreements (IGA) approved by City Council since the inception of the TIF program. This dataset does not include public Infrastructure projects.

Chicago Microlending Institute (CMI) (City of Chicago, 2020) consists of a dataset reflecting the lender, location, business industry, and borrower demographics for small businesses supported by the City's revolving loan fund.

Ordinance Violations (City of Chicago, 2020) is the dataset made of a list of ordinance violations filed with the Department of Administrative Hearings. This data set reflects violations brought before the Chicago Department of Administrative Hearings. It does not reflect violations brought before the Circuit Court of Cook County. Each row of data represents a unique violation. Multiple

violations may be associated with a single case. This dataset currently lists violations issued by the Department of Buildings.

General Fertility Rates in Chicago, by Year, 1999-2009 (City of Chicago, 2020) consists of a dataset containing the annual general fertility rate (births per 1,000 females aged 15-44 years) with corresponding 95% confidence intervals, by Chicago community area, for the years 1999 – 2009. To keep data consistent with the percentage rate, the dataset is modified to present the annual general fertility rate as births per 100 females aged 15-44 years.

Gonorrhea Cases for Females Aged 15 - 44 in Chicago, by Year, 2000 - 2014 (City of Chicago, 2020) and Gonorrhea Cases for Males Aged 15-44 in Chicago, by Year, 2000 - 2014 (City of Chicago, 2020) are two datasets merged together to for one column “gonorrhea_percent” in “datum’” principal data frame. The former dataset contains the annual number of laboratory-confirmed cases of gonorrhea (Neisseria gonorrhoeae) among females aged 15-44 years and annual gonorrhea incidence rate (cases per 100,000 females aged 15-44 years) with corresponding 95% confidence intervals by Chicago community area, for years 2000 – 2014. The later dataset contains the annual number of laboratory-confirmed cases of gonorrhea (Neisseria gonorrhoeae) among males aged 15-44 years and annual gonorrhea incidence rate (cases per 100,000 males aged 15-44 years) with corresponding 95% confidence intervals by Chicago community area, for years 2000 – 2014.

Chlamydia Cases Among Females Aged 15-44 in Chicago, by Year, 2000-2014 (City of Chicago, 2020) and Chlamydia Cases Among Males Aged 15-44 in Chicago, by Year, 2000-2014 (City of Chicago, 2020) are two datasets merged together to for one column “chlamydia_percent” in “datum’” principal data frame. The former comprises the annual number of laboratory-confirmed cases of chlamydia (Chlamydia trachomatis) among females aged 15-44 years and annual chlamydia incidence rate (cases per 100,000 females aged 15-44 years) with corresponding 95% confidence intervals by Chicago community area, for years 2000 – 2014, while

the later contains annual number of laboratory-confirmed cases of chlamydia (Chlamydia trachomatis) among males aged 15-44 years and annual chlamydia incidence rate (cases per 100,000 males aged 15-44 years) with corresponding 95% confidence intervals by Chicago community area, for years 2000 – 2014.

Preterm Births in Chicago, by Year, 1999 – 2009 (City of Chicago, 2020) is a dataset that contains the annual number of preterm births and the percent of total births these preterm births represent, with corresponding 95% confidence intervals, by Chicago community area, for the years 1999 – 2009.

Low Birth Weight in Chicago, by Year, 1999 – 2009 (City of Chicago, 2020) is the dataset that contains the annual number of low birth weight births and the percent of total births these low birth weight births represent, with corresponding 95% confidence intervals, by Chicago community area, for the years 1999 – 2009.

Life Expectancy (City of Chicago, 2020) consists of the dataset providing the average life expectancy and corresponding confidence intervals for each Chicago community area for the years 1990, 2000 and 2010.

Tuberculosis Cases and Average Annual Incidence Rate, Chicago, 2007- 2011 (City of Chicago, 2020) is the dataset composed of the annual number of cases of tuberculosis and average annual tuberculosis incidence rate (new cases per 100,000 residents) with corresponding 95% confidence intervals, by Chicago community area, for the years 2007 – 2011.

Births to Mothers Aged 15-19 Years Old in Chicago, by year, 1999-2009 (City of Chicago, 2020) is the dataset containing the annual number of births to mothers aged 15-19 years old and annual birth rate (births per 1,000 females aged 15-19

years) with corresponding 95% confidence intervals, by Chicago community area, for the years 1999 – 2009.

Infant Mortality in Chicago, 2005– 2009 (City of Chicago) is the dataset that contains the annual number of infant deaths annually, cumulative number of infant deaths, and average annual infant mortality rate with corresponding 95% confidence intervals, by Chicago community area, for the years 2005 – 2009.

Screening for Elevated Blood Lead Levels in Children Aged 0-6 Years by Year, Chicago, 1999 - 2013 (City of Chicago, 2020) is the dataset containing the annual number and estimated rate per 1,000 children aged 0-6 years receiving a blood lead level test, and the annual number and estimated percentage of those tested found to have an elevated blood lead level, with corresponding 95% confidence intervals, by Chicago community area, for the years 1999 – 2013.

Selected Public Health Indicators by Chicago Community Area (Chicago, 2020) is the dataset that contains a selection of 27 indicators of public health significance by Chicago community area, with the most updated information available. The indicators are rates, percents, or other measures related to natality, mortality, infectious disease, lead poisoning, and economic status. For the purpose of this study, 13 indicators are selected and shown in Table 2.5.

Diabetes Hospitalizations in Chicago, 2000 - 2011 (City of Chicago, 2020) consists of a dataset that contains the annual number of hospital discharges, crude hospitalization rates with corresponding 95% confidence intervals, and age-adjusted hospitalization rates with corresponding 95% confidence intervals, for the years 2000 – 2011, by Chicago U.S. Postal Service ZIP code or ZIP code aggregate.

Asthma Hospitalizations in Chicago by Year 2000 - 2011 (City of Chicago, 2020) is a dataset containing the annual number of hospital discharges, crude hospitalization

Table 2.5. *Selected public health indicators by Chicago Community Area*

Column Name	Data Type	Measure Description
Birth rate	float	Births per 1000
Prenatal care beginning in the first semester	float	Percent of females delivering a live birth
Assault (Homicide)	float	Assault (Homicide) per 100,000 persons (age adjusted)
Cancer (All sites)	float	Cancer (All sites) per 100,000 persons (age adjusted)
Diabetes-related	float	Diabetes-related per 100,000 persons (age adjusted)
Firearm-related	float	Firearm-related per 100,000 persons (age adjusted)
Stroke (Cerebrovascular disease)	float	Stroke (Cerebrovascular disease) per 100,000 persons (age adjusted)
Childhood blood level screening	float	Childhood blood level screening per 1000 children aged 0-6 years old
Below poverty level	float	Percent of households below poverty level
Crowded housing	float	Percent of houses classified as crowded
Dependency	float	Dependency percent of persons aged less than 16 or over 64 years
No high school diploma	float	No high school diploma percent of persons aged 25 and older
Per capita income	float	Per capita income by 2011 inflation adjusted dollars

rates with corresponding 95% confidence intervals, and age-adjusted hospitalization rates (per 10,000 children and adults aged 5 to 64 years) with corresponding 95% confidence intervals, for the years 2000 – 2011, by Chicago U.S. Postal Service ZIP code or ZIP code aggregate. Clinical Care, Mortality, Morbidity, Physical Environment and Health Behaviors (City of Chicago, 2020) is a an aggregated data frame that updates the principal data frame "datum" with data from the data frame "addendum" about Clinical Care,

Mortality, Morbidity, Physical Environment and Health Behaviors. For details about the data type and data source, see addendum data frame above.

Selected Socioeconomic Indicators in Chicago, 2008 – 2012 (City of Chicago, 2020) is the dataset containing a selection of six socioeconomic indicators of public health significance and a “hardship index,” by Chicago community area, for the years 2008 – 2012. For the purpose of this study, the indicators taken into consideration are: PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA, which is the percent of persons over the age of 25 years without a high school education, and HARDSHIP INDEX, which is the score that incorporates each of the six selected socioeconomic indicators (see dataset description).

Street Lights - All Out (City of Chicago, 2020) is the dataset that contains all open reports of "Street Lights - All Out" (an outage of 3 or more lights) made to 311 and all requests completed since January 1, 2011.

Tree Debris (City of Chicago, 2020) is the dataset that contains all open tree debris removal requests made to 311 and all requests completed since January 1, 2011.

Alley Lights Out (City of Chicago, 2020) is the dataset containing all open 311 reports of one or more lights out on a wooden pole in the alley and all completed requests since January 1, 2011.

Potholes (City of Chicago, 2020) is the dataset containing all open pot hole requests and all completed requests since January 1, 2011.

Non-Hispanic African American or Black Population (Zip Atlas, 2020) is the dataset used to update the principal data frame with a column holding data on the percentage of Non-Hispanic African-American or Black among the total population of a given zip code. As for the Non-Hispanic White Population (Zip Atlas, 2020) is that dataset that does the update to "datum' with the percentage of Non-Hispanic White among the total population per zip code area. To obtain data on the Hispanic or Latino Population, the Chicago Health Atlas is utilized (Chicago Health Atlas, 2020) for the percentage of Hispanics or Latinos among the total population of a given Chicago's zip code area. Respective datasets about people of Chinese, Korean and Indian descent function in identical roles.

3. METHODOLOGY

3.1. Data Garnering

In order to collect data, a number of Python's programming techniques are used depending on the task at hand and the source of data. The Jupyter Notebook -an open-source web application that allows the creation and sharing of documents that contain live code, equations, visualizations and narrative text is used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization and machine learning

The *download()* method of the *wget* library is utilized to download online data available in the comma separated values (csv) format. Thanks to this method, during this study it was possible to download data about the target variable, CPS, Covid-19, testing sites, traffic crashes, problem landlord buildings , crimes, public libraries, affordable rental housing, grocery stores, MMRP, Tax Increment Financing

(TIF), Chicago Microlending Institute(CMI), ordinance Violation, general fertility rates in Chicago, gonorrhea cases for females Aged 15 - 44 in Chicago, gonorrhea cases for males aged 15-44 in Chicago, chlamydia cases among females aged 15-44 in Chicago, chlamydia cases among males aged 15-44 in Chicago, preterm births, low birth weight, life expectancy, tuberculosis, births to mothers aged between 15 and 19, Infant mortality, childhood lead poisoning, selected health indicators, diabetes hospitalizations, asthma hospitalizations, selected socio-economic, all-lights out, tree debris, alley lights, and potholes.

The *get()* method of the "requests" library is used for web scraping and importing data in the text format. Thanks to this method, the data about , *inter alia*, community areas, clinics, primary care facilities and demography are accessed.

As for data that is available in the format of *json*, the *read()* method of the "json" library was used to retrieve data into the Jupyter Notebook. Henceforth, it was possible to obtain the CPD, addendum, CFD, traffic crashes and problem landlord buildings data from a json file. This step is followed by data organization explained in details in later paragraphs.

A considerable amount of geographical data is only available for retrieval through foursquare.com IDE. For that task, it is necessary to set up the client ID and the client secret to authenticate the request to foursquare.com. Those authentifications are paired with parameters such as the latitude, longitude, version, query, radius, the uniform resource locator (url) and limit to complete the request to foursquare.com. It is through that pathway that a part of data about hospitals in the City of Chicago is retrieved.

In the case of datasets already available on the user's computer under the csv format, the *read_csv()* method of the *"pandas"* library is used for reading the data into the pandas' data frame. Similarly, the *read_json()* method is used to import data in the format of json.

The retrieval of data about nursing homes in Chicago requires a mix of techniques, among others, web scraping, text editing, list comprehension, csv file manipulation, and lxml file editing. This complexity calls for the creation of the "NursingHome" class. The *"NursingHome"* class features "homeList" that takes in the argument as an url connecting to the website containing the nursing homes and returns a list containing names of nursing homes and their respective addresses. The Nurse_pd() method of “NursingHome” class takes in five arguments all of them being lists containing names of nursing homes and their respective addresses to return a pandas dataframe containing names of nursing homes and their respective addresses.

3.2. Data Processing

Prior to the organization of the principal pandas dataframe *“datum”* and the final dataset for machine learning training, a number of steps has to be taken to first process the collected data.

The *"zipchi"* pandas dataframe is first created to help in preprocessing data for the target variable -”contagion intensity”- according to rules of the Susceptibility, Infection, Recovery (SIR) model. The *”zipchi”* data frame is made of columns of *'zip_code', 'week_number', 'week_start', 'week_end', 'cases_weekly', 'cases_cumulative', 'case_rate_weekly', 'case_rate_cumulative', 'tests_weekly', 'tests_cumulative', 'test_rate_weekly', 'test_rate_cumulative', 'percent_tested_positive_weekly', 'percent_tested_positive_cumulative', 'deaths_weekly', 'deaths_cumulative', 'death_rate_weekly', 'death_rate_cumulative', 'population', 'row_id', 'zip_code_location', 'transmission_rate', 'susceptible_population', 'suscebility_fraction', 'infection_fraction',* and *'weekly_spread_rate'.*

The *“zipcode”* data frame is used to retrieve and process the zip code areas in Chicago, which later on serve as the index column for the principal data frame.

The *"zipcov"* pandas dataframe is then created to later on be transformed into the "datum" pandas dataframe. This prerunner of the principal data frame is made of 17 columns, namely; *"last_day", "first_day", "positive_total", "positive_rate", "positive_percent", "test_total", "test_rate", "test_percent", "positive_test_rate", "positive_test_percent", "death_total", "death_rate", "death_percent", "death_positive_percent", "population", "number_days", "latitude",* and *"longitude"*. Except for the *"population"* column, or other columns do not take place in the neural network training because they are specific to COVID-19 properties only available once the disease is pandemic.

For data processing, the *wrangler()* function takes in three arguments consisting of the number of a given one zip code area, a pandas dataframe from which to retrieve data related to COVID-19 in Chicago and a pandas dataframe from which to obtain indexes and columns to serve in the principal pandas dataframe to finally return a pandas dataframe to feed initial data to the principal pandas dataframe.

The *communityBased()* function on its turn takes in seven arguments, namely; a pandas dataframe to update, a pandas dataframe with data based on community area, a name of the new column to be created and joined into "*datumx*", a counter of the relevant community areas included in the zipcode, sum of the data points to be counted in the zipcode, column containing the the name of the community area, column from which to retrieve the data points and returns an updated pandas dataframe to feed initial data to the principal pandas dataframe

Another function, *latLong()* takes in a pandas dataframe to update that contains columns for street address, city and state and returns an updated pandas dataframe with new columns containing latitude and longitude obtained from foursquare.com.

The *numZipPercent()* function takes in seven argument, namely; the principal pandas dataframe to update, a pandas dataframe from which to obtain zip codes

and names, column name to be used to count the number of facilities in a given zip code, column name to be used to store names of facilities in a given zip code, column name to be used to to store the number of facilities per 100 residents a given zip code, column name for the column in dfx containing zip code numbers, column name for the column in dfx containing names of facilities to eventually return an updated pandas dataframe with new columns containing number, names and percentage of facilities in a zip code area.

The function of *zipBased()* takes in seven *a*rguments, which are the principal pandas dataframe to be updated, a vassal pandas dataframe from which to retrieve data, name of the new column in the principal pandas dataframe to be updated, name of the counter temporary column, name of the summation temporary column, name of the zipcode column, and the name of the column of the assal pandas dataframe from which to retrieve data. *zipBased()* then returns an updated principal pandas dataframe.

In the aftermath of *requests.get,* the file obtained in *txt* format is then passed to *BeautifulSoup()* method for parsing as an *xml* file or any other markup language supported by that method.

3.3. Data Organization

In the wake of data collection and data processing, data organization starts with the creation of a pandas dataframe shell *"datum",* which becomes the principal data frame where to store all of the data that will be needed for the prediction of the COVID-19 contagion intensity per zip code.

In *"datum",* rows are Chicago's zip code areas, i.e. serving as indexes. As Chicago has 59 zipcodes, "datum" has consequently the same number of rows. At the end, "*datum*" is made of 175 columns, of which two of them are of datetime datatype, 117 are of float64 data type, 30 are of int64 data type, and 26 are of object. The last

are those columns made of lists. The last column of “datum” is contagion intensity, which is to serve as the target variable during the supervised neural network training.

Nonetheless, not all of the “datum” data is going to be used for neural network training. The *“datum1”* data frame contains the final data to be utilized in the training of the neural network. The row of “60666” is dropped because it is the zip code area of O’Hare International Airport and there is no resident of Chicago registered there as her/his abode place. The number of columns left is 85 (see the attached covid_19_chicago.ipynb for details).

For the neural network, the data is fetched from the pandas dataframe into a numpy array of shape 58 by 84, i.e. minus the target variable column.

3.4. The Target Variable

The COVID-19 contagion intensity serves as the target variable for the training of the neural network model. Each Chicago’s zipcode zipcode has the COVID-19 contagion intensity calculated based on the modified SIR model.

According to Wikipedia's webpage of Compartmental models in epidemiology (Wikipedia, 2020), the SIR stands for Susceptible, Infectious and Removed.

In the SIR model, susceptible is the number of susceptible individuals. When a susceptible and an infectious individual comes into "infectious contact", the susceptible individual contracts the disease and transitions to the infectious compartment. Infectious is the number of infectious individuals. These are individuals who have been infected and are capable of infecting susceptible individuals. Lastly, removed is the number of removed (and immune) or deceased individuals. These are individuals who have been infected and have either recovered from the disease and entered the removed compartment, or died. It is

assumed that the number of deaths is negligible with respect to the total population. This compartment may also be called "recovered" or "resistant". Susceptible, infectious and removed are ultimately the composing variables of the SIR model. The SIR model prediction is utilized to calculate the rate of contagion or the spread rate of infectious diseases that are transmitted from human to human in a given area during a certain amount of time.

S, I, and R variables depict the number of people in each compartment at a given time. The representation of the number of susceptible, infectious and removed individuals is likely to vary over time. Due to the lack of key epidemiological metrics -such as reproduction number and recovery rate-, and also due to the lack of the availability of mobility data, the target variable -in this study-, is the result of an average of the weekly probabilities calculated over the period starting from the week when the first positive case was reported in a given zip code area.

In this study, the SIR model that is taken into consideration is the SIR with dynamics of the disease in one region and constant population. The dynamics of COVID-19 are considered because it is a communicable disease that is spreading. Despite the lack of some dynamics, the SIR model in this project is thought to represent at certain degree the mass-action transmission properties of COVID-19. The contagion intensity in this project is taken as the average number of infections caused by a single infectious subject in a wholly susceptible population. That modified form might be less than, more than or equal to the actual version.

Therefore, the lack of data on mobility between Chicago zip code areas and other parts, and given that the aim is to demonstrate how to utilize the preexisting data on health and socio_economics to predict the coronaviral contagion intensity at its outset within a given Chicago's zip code area, the classical SIR model undergoes a transformation to obtain an oversimplified SIR Equation (1) to compute the COVID-19 contagion intensity.

$$h_{j,t} = \frac{\beta_t S_{j,t}\left(1 - e^{\left(-\sum_k m^t_{j,k} X_{k,t} Y_{j,t}\right)}\right)}{1 + \beta_t Y_{j,t}} \quad (1)$$

Where:

- j denotes a location. This time a Chicago's zip code area
- t denotes a given week during the outbreak
- $h_{j,t}$ denotes COVID-19 contagion intensity in Chicago's zip code area *j* during the *w*
- β_t is the transmission rate in week *t*
- $m^t_{j,k}$ reflects mobility from Chicago's zip code area *k* to Chicago's zip code area *j* during week *t*. In the modified SIR model this value is 1
- $X_{k,t}$ denotes the fraction of the infected populations in week t at location k. It is given by:

$$X_{k,t} = \frac{I_{k,t}}{N_k} \quad (2)$$

- $Y_{j,t}$ denotes the fraction of the susceptible populations in week t in Chicago's zip code area *j*. It is given by:

$$Y_{k,t} = \frac{S_{j,t}}{N_j} \quad (3)$$

- N_j is the population sizes in Chicago's zip code area j
- N_k is the population sizes in Chicago's zip code area k
- $I_{j,t+1}$ is a Bernoulli random variable with probability h(t,j). It is given by:

$$I_{j,t+1} = I_{j,t} - \frac{\beta_{j,t} S_{j,t} I_{j,t}}{N_j} - \frac{\alpha S_{j,t} \sum_k m^t_{j,k} X_{k,t} \beta_{k,t}}{N_j \sum_k m^t_{j,k}} - \gamma I_{j,t} \quad (4)$$

- $S_{j,\,t+1}$ is the number of susceptible individuals in week t+1 at Chicago's zip code area j. It is given by:

$$S_{j,\,t+1} = S_{j,\,t} - \frac{\beta_{j,\,t} S_{j,\,t} I_{j,\,t}}{N_j} - \frac{\alpha S_{j,\,t} \sum_k m^t_{j,\,k} X_{k,\,t} \beta_{k,\,t}}{N_j \sum_k m^t_{j,\,k}} - \gamma I_{j,\,t} \qquad (5)$$

- $R_{j,\,t+1}$ is the basic reproduction number in week t+1 at location j. It is given by:

$$R_{j,\,t+1} = R_{j,\,t} + \gamma I_{j,\,t} \qquad (6)$$

- γ is the recovery rate
- β is a coefficient denoting the modal share or the intensity of mobility.

The lack of data on reproduction rate, recovery rate, mobility and intensity of mobility, the classical SIR model undergoes a transformation to obtain an oversimplified SIR equation.

Equation (1) becomes:

$$h_{j,t} = \frac{\beta_t S_{j,\,t}\, 1 - e^{(-\sum_j Y_{j,\,t})}}{1 + \beta_t Y_{j,\,t}} \qquad (7)$$

3.5. Exploratory Data Analysis (EDA)

For EDA, in this study both descriptive statistics and data visualization techniques are the fundamental tools used to carry out analytic exploration of the dataset. In addition, this EDA consists of the examination of the relationship between the feature variables and the target variable. The outcome of EDA leads to the insight into the underlying distributions, into the realization of the best placed activation functions for the neural network and to the intuition about how to interpret the results of the predictive models to be developed later on.

Given the fact that data points were not collected through means of a statistical random sampling, and for the purposes of prediction accuracy, neither the target variable nor the feature variables are transformed into standard normal distribution. This approach contributes to the generation of a model that would be at least close to the facts presented by the real world.

3.5.1. Descriptive Statistics

For the descriptive statistics, the analysis focuses on the central tendency, the measure of dispersion and the distribution of data.

In this part, the analysis focuses on the centrality, spread, quantiles, compactness, covariance and collinearity; both of the dependent variables and the target variable, as well as - if applicable-, between the feature variables and the independent variable.

The *describe()* method of the pandas dataframe provides a shortcut for the calculation of the mean, maximum, minimum, quartiles and standard deviation (STD) of every numeric column of the "*datum1*" dataset.

The resulting pandas dataframe is assigned into a pandas data frame called *cent_df*. The data frame cent_df is then updated with the mode, median, variance, mean absolute deviation (MAD), kurtosis, skewness and the mean of kernel density estimation (kde).

The *mode()* method of the pandas dataframe yields the value of the most occuring entry along the columns. On its turn, the *median()* method gives the data point placed midway. As of the *var()* method of the pandas dataframe, it provides the variance as the measure of the spread of the data identified as feature or target variables. The *mad()* method of the pandas dataframe computes the mean absolute

deviation (MAD), which is a means to gauge the average distance between each data value and the mean.

The kurtosis will help to measure the outliers present in the distribution of the target variable dataset. At the other hand, the skewness will gauge the degree of distortion from the symmetrical bell curve or the normal distribution. The kernel density estimation (KDE) is used to estimate the probability density function of the target variable and helps to visualize the distribution of the target variable.

$$KDE_j = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h\sqrt{2\pi}} e^{-0.5(\frac{X_j - X_i}{h})^2} \quad (8)$$

Where:

- X_i is the ith datapoint
- X_j is the jth datapoint on which the calculation of KDE is performed
- n is the total number of data points
- KDE_j is the kernel density estimation of the jth datapoint
- h is the bandwidth

With the *cov()* method of the pandas dataframe, the covariance is computed to provide an insight into how each feature variable is related to the target variable. For the correlation, the Pearson Correlation Coefficient, r, is to be used in order to gauge the strength and the direction of the linear relationship between a given feature variable and the target variable.

3.5.2. Data Visualization

The map of chicago is used to obtain visual comparative insight into the gravity of COVID-19 contagion intensity per zip code area. The map shows the COVID-19 intensity of contagion per zip code area in Chicago. The higher the COVID-19

contagion intensity per zip code, the bolder is the blue-purple color in that zip code area.

As an integral part of EDA, data visualization is applied to confer visual insights into the relationship between the feature variables and the target variable. The feature variables are divided into groups of Feature Variables Specific to COVID-19, Infrastructure Investment, Government Response, Capital Investment, Safety, Health Indicators, Demography and Socio-Economic. The category of health indicators is made of subcategories of clinical care, mortality, morbidity and health behavior.

The function *plotter()* that takes in three arguments, namely; columns against which to plot with the target variable, pandas dataframe from which to retrieve data points and name of the plot. The function *plotter()* uses *numpy*'s method *polyfit()* to fit data on a linear regression and used *scatter()* and *line()* methods of *bokeh.plotting.figure()* class to return subplots showing data point to data point scatter plot and the best linear regression.

The set of infrastructure investment feature variables is composed of Chicago Police Department, fire stations, hospitals, clinics, primary care dispensaries, COVID-19 test sites, public schools, nursing homes, and libraries. The set of government response feature variables comprises the measurement of aspects that require the government intervention for the sustainability of the wellbeing of the residents. This set comprises affordable rental housing, childhood poisoning level, complaints about non functioning street lights, complaints about non functioning alley lights, presence of tree debris and potholes in streets. Capital investment set contains data about microfinance lending, tax increment financing, micro-market recovery program and groceries. The collection of safety features englobes traffic accidents per 100 residents, problematic landlords 100 residents, crimes per 100 residents, violations per 100 residents, assault and homicide per 100 residents, and firearm related crimes per 100 residents. The ensemble of

health indicators contains four subcategories, which are clinical care, mortality, morbidity and health behavior.

The set of socio-economic features is made of 9 elements, namely; the percentage of illegitimate police stops, the percentage of people on social support, the percentage of unemployed people, the percentage of of people without high school diploma, per capita income, the percentage of people living below poverty level, the number crowded housing per 100 residents, the percentage of people in dependency, and hardship_index. The correlation between the socio-economic features and the COVID-19 contagion intensity corresponds to the observation fitting with the rationale going like "the better the socio-economic conditions, the lesser the intensity of COVID-19 contagion".

This set of demography is an ensemble composed of the race-ethnicity designation of peoples groups, namely; African-American or Black people percentage per zip code, Non-Hispanic White people percentage per zip code, Hispanic or Latino people percentage per zip code, Chinese people percentage per zip code, Korean people percentage per zip code, and Asian Indian people percentage per zip code.

3.6. Machine Learning: Neural Network Model

The model used for training the dataset is the artificial neural network. As the data points of the contagion intensity can take any non-negative value; therefore, the target variable consists of continuous variables rather than binary or logistic variables. Hence, the prediction of the output layer consisting of polynomial regression.

3.6.1. Dataset Development for the Neural Network Model

The function *featureName()* sets columns to be used as features in the training of the neural network by taking in an arbitrary tuple containing a list or strings of the

names of the columns of "*datum1*" dataframe serving as features. The function featureName returns a list containing names of the columns serving as a feature dataset of the training model.

The function of *featureData()* retrieves data points for the feature dataset by taking in two arguments. One is a pandas dataframe from which to retrieve data points to serve as an input array to the neural network. The second is a list containing names of column names from the data frame from which to retrieve data points. The function of *featureData()* then returns a numpy vector or matrix containing data points to serve as an input array to the neural network model.

3.6.2. Structure of the Neural Network Model

The neural network model is made of two parts. For the first part of the neural network, it is dynamic at taking any number layers. The activation function for the hidden layers is the leaky ReLU and for the outer layer is ReLU. The second part of the neural network takes in the parameters yielded by the first part of the neural network. This part is flexible at taking any number of layers. The activation function for hidden layers is Softplus and for the output layer is Swish.

Before shifting from the first part of the neural network to the second part of the neural network, the back propagation is performed in order to learn parameters to pass to the second neural network. On its turn, the second part of the neural network performs the forward propagation and the backpropagation to give the final parameters of the neural network model.

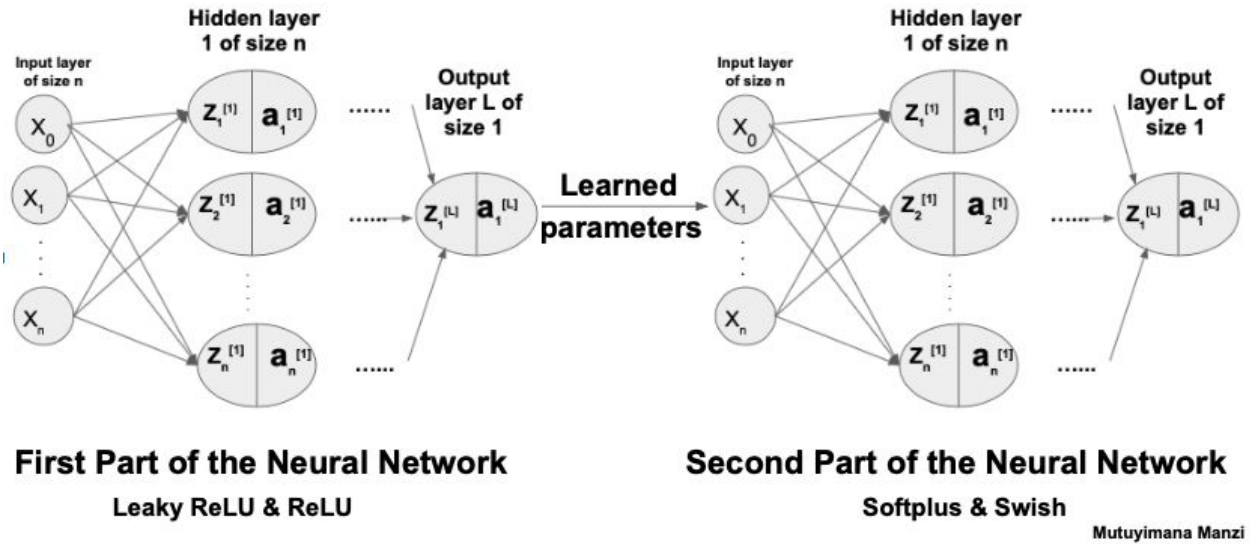
3.6.3. Visual and Mathematical Illustration of the Neural Network Model

The number of nodes in each layer are determined by the function *nnStructure* Figure 1. visualizes the forward propagation of each part of the neural network.

This illustration does not include the back propagation of either part of the neural network.

The function *nnStructure()* determines the number of nodes in each layer of the neural network. The function *nnStructure()* takes in three arguments. The first argument is the size of the input layer. The second argument is the size of the output layer and the third argument is the number of layers. Then the function *nnStructure()* returns a list whose size equals the number of layers in the neural network and whose elements are numbers equaling the number of nodes in each corresponding layer.

Figure 1. *Illustration of the neural network model*



The mathematical illustration of the neural network model is illustrated by Equation (9) and Equation (10).

In Equation (9) and Equation (10), it shows that the neural network model has parameters, which are weights and biases. In this study, the neural network model learns parameters to achieve predictions without overfitting or underfitting.

Depending on the structure of the neural network, the function *parameterInitialization()* randomly initializes the parameters to be used to train the neural network by taking in one argument that is a list of nodes per each layer of the neural network, and then returning a dictionary containing matrices of weights W and biases B for each layer of the neural network. Later on those parameters are updated.

$$First - Neural - Network : \begin{cases} Input - Layer : \begin{cases} z^{[1](i)} = W^{[1]} x^{(i)} + b^{[1]} \\ a^{[1](i)} = leakyrelu(z^{[1](i)}) \end{cases} \\ Output - Layer : \begin{cases} z^{[L](i)} = W^{[L]} a^{[L-1](i)} + b^{[L]} \\ \hat{y}^{(i)} = a^{[L](i)} = relu(z^{[L](i)}) \end{cases} \end{cases} \tag{9}$$

$$Second - Neural - Network : \begin{cases} Input - Layer : \begin{cases} z^{[1](i)} = W^{[1]} x^{(i)} + b^{[1]} \\ a^{[1](i)} = softplus(z^{[1](i)}) \end{cases} \\ Output - Layer : \begin{cases} z^{[L](i)} = W^{[L]} a^{[L-1](i)} + b^{[L]} \\ \hat{y}^{(i)} = a^{[L](i)} = swish(z^{[L](i)}) \end{cases} \end{cases} \tag{10}$$

Where:

- $X^{(i)}$ is the training example the training example i
- $W^{[l]}$ is the matrix of parameters associated with layer l
- $b^{[l]}$ is the bias vector associated with layer l
- $z^{[l](i)}$ is the vector of layer *l* resulting from matrix multiplication of $W^{(l)}$ and $X^{(i)}$ or $a^{[l-1](i)} + b^{[l]}$
- $a^{[l](i)}$ is the vector resulting from the activation of $z^{[]()}$
- $\hat{y}^{(i)}$ is the same as $a^{[L]()}$ which is the output of the neural network -either first or second one- for example

3.6.4. Activation Functions

Skewness values of the target variable and the feature variables are greater than one in most of the cases, i.e. there is an overall perverence of the dataset being positively skewed. And most of the data points are concentrated on the left side of

the distribution curve.Therefore, the activation functions used to train the neural network model are those functions that would perform an increasing effect of the correlation for the data points on the lower end of the distribution. Consequently, the leaky Rectified Linear Units (Leaky ReLU), Rectified Linear Units (RelU), Swish and Softplus functions are, among others, some of the candidates. Leaky ReLU and ReLU are used in the first part of the neural network. The former serves as the activation function for the hidden layers of the neural network, while the latter's role consists of the activation function of the output layer.

Swish function and Softplus function perform in the second part of the neural network; with the former being the output layer activation function and the later as the activation function for the hidden layers of the neural network. However, due to the fact that the domain of definition of both the Swish and Softplus functions does not reach high values of the independent variable, leaky ReLU and leaky ReLU functions are used to preprocess the first part of the neural network.

3.6.4.1. Swish Function

The function *swish()* implements Equation (11), which is the Swish function. The function *swish()* takes in one argument in the form of a matrix or a vector and returns a matrix or vector serving as the output of the outer layer in the forward propagation phase of the second part of the neural network model.

$$f(x) = \frac{x}{1 + e^{-\beta x}} \qquad (11)$$

Where:

- x is the independent variable
- β is a constant. β equals 0.5 in this study

3.6.4.2. Softplus Function

The softplus function given by Equation (12) is implemented by the function *softplus()* in the programm. The function *softplus()* takes in one argument in the form of a matrix or a vector and returns a matrix or vector serving as the output of the inner layers in the forward propagation phase of the second part of the neural network model.

$$f(x) = ln(1 + e^x) \qquad (12)$$

Where:

- x is the independent variable

3.6.4.3. Rectified Linear Unit (ReLU) Function

In the code, the ReLU function by Equation (13) is implemented through the function *relu(),* which takes in one argument in the form of a matrix or a vector to finally return a matrix or vector serving as the output of the outer layer in the forward propagation phase of the first part of the neural network model.

$$f(x) = max(1, x) \qquad (13)$$

Where:

- x is the independent variable

3.6.4.4. Leaky Rectified Linear Unit (leaky ReLU) Function

The leaky ReLU function given by Equation (14) is implemented by the function *leakyRelu()* in the programm. The function *leakyRelu() takes* in one argument in the form of a matrix or a vector and returns a matrix or vector serving as the output of

the inner layers in the forward propagation phase of the first part of the neural network model.

$$f(x) = max(0.01x, x) \qquad (14)$$

Where:

- x is the independent variable

3.6.5. Forward Propagation

The step of forward propagation of the neural network model serves to calculate the equations (9) and (10). That task is implemented through the function *forProp(),* three arguments. Those three arguments are an input numpy array of shape (number of features, number of training examples), a python dictionary containing parameters, and a string indicating which part of the neural network to perform based on the name of the output layer activation function. The function *forProp()* returns a dictionary containing matrices of linearization Z and of activation A of all layers.

3.6.6. Cost Function : The Mean Squared Error(MSE)

$$L(A, Y) = \frac{1}{2m} \sum_{i=1}^{m} (a^{[L](i)} - y^{(i)})^2 \qquad (15)$$

Where:

- $a^{[L](i)}$ is the prediction for the training example *i* in the last layer L.
- $y^{(i)}$ is the actual value of the target variable for the training example *i*.
- *m* is the number of the training examples.

The mean squared error (MSE) is used as the loss function, whereby the target variable (contagion intensity) is subtracted from the predictions yielded by the output of the last layer of the neural network. The function *lossFunction()* implements Equation (14) of MSE by taking in two arguments. The first argument is the output vector of the neural network (predictions). The second argument is a vector that is the target variable (contagion intensity). The function *lossFunction()* then returns a scalar that is the loss between the output of the neural network and the target variable.

3.6.7. Gradient Descent

The computation of the gradient descent makes use of the calculation of the derivatives of the linearization functions, weights and biases.

3.6.7.1. Derivatives in the First Part of the Neural Network Model

In this calculation phase of the neural network, the activation function of the hidden layers is the leaky ReLU function, while the activation function of the output layer is the ReLU function.

$$dZ^{[L]} = \frac{dLo}{dZ^{[L]}} = \frac{dLo}{dA^{[L]}} \frac{dA^{[L]}}{dZ^{[L]}} \tag{16}$$

$$\frac{dLo}{dZ^{[L]}} = \frac{d}{dA^{[L]}}\left(\frac{1}{2m}(A - Y)^2\right) = \frac{1}{m}(A - Y)^2 \tag{17}$$

$$\frac{dA^{[L]}}{dZ^{[L]}} = \frac{d}{dZ^{[L]}}\left(max(1, A^{[L](i)})\right) = \begin{cases} 1 & \text{if } Z^{[L](i)} \geq 0 \\ 0 & \text{if } Z^{[L](i)} < 0 \end{cases} \tag{18}$$

$$dZ^{[L]} = \frac{dLo}{dZ^{[L]}} = \frac{dLo}{dA^{[L]}} \frac{dA^{[L]}}{dZ^{[L]}} = \begin{cases} \frac{1}{m}(A - Y) & \text{if } Z^{[L](i)} \geq 0 \\ 0 & \text{if } Z^{[L](i)} < 0 \end{cases} \tag{19}$$

$$dW^{[l]} = \frac{1}{m} dZ^{[L]} A^{[l-1]^T} \tag{20}$$

$$db^{[l]} = dZ^{[L]} \tag{21}$$

$$dZ^{[l]} = W^{[l+1]^T} dZ^{[L+1]} g^{[l]'}(Z^{[l]}) \tag{22}$$

$$g^{[l]'}(Z^{[l]}) = \begin{cases} 1 & \text{if } Z^{[L](i)} \geq 0 \\ 0.01 & \text{if } Z^{[L](i)} < 0 \end{cases} \tag{23}$$

Where:

- $[L]$ denotes the output layer of the neural network
- $[l]$ denotes any layer, other than the output layer of the neural network
- i denotes i^{th} training example
- m is the number of the training examples
- $Z^{[L]}$ is the linearization in the output layer
- $A^{[L]}$ is the linearization in the output layer
- Lo is the loss function or cost function
- $\frac{dLo}{A^{[L]}}$ is the derivative of the cost function with respect to $A^{[L]}$
- $dZ^{[L]} = \frac{dLo}{dZ^{[L]}}$ is the derivative of the loss function with respect to $Z^{[L]}$
- $dZ^{[l]} = \frac{dA^{[l]}}{dZ^{[l]}}$ is derivative of the activation function $A^{[l]}$ with respect to $Z^{[L]}$
- $dW^{[l]}$ is the derivative of the weight matrix W in the layer l
- $db^{[l]}$ is the derivative of the bias vector b in the layer l
- $g^{[l]'}(Z^{[l]})$ is the derivative of the activation function in layer l with respect to $Z^{[l]}$

3.6.7.2. Derivatives in the Second Part of the Neural Network Model

In this part of the neural network, the activation function of the hidden layers is the $Softplus$ function, while the activation function of the output layer is the $Swish$ function. The formula of equations (16), (15), (20), (21) and (22) in the first part of the neural network model are the same for the second part of the neural network model.

$$\frac{dA^{[L]}}{dZ^{[L]}} = \frac{d}{dZ^{[L]}}\left(\frac{Z^{[L]}}{1-e^{-0.5Z^{[L]}}}\right) = \frac{1-e^{-0.5Z^{[L]}}+0.5\ 1-e^{-0.5Z^{[L]}}Z^{[L]}}{(1-e^{-0.5Z^{[L]}})^2} \tag{24}$$

$$dZ^{[L]} = \frac{dLo}{dZ^{[L]}} = \frac{dLo}{dA^{[L]}}\frac{dA^{[L]}}{dZ^{[L]}} = \frac{1}{m}(A - Y)\left(\frac{1+e^{-0.5Z^{[L]}}+0.5\ e^{-0.5Z^{[L]}}Z^{[L]}}{(1-e^{-0.5Z^{[L]}})^2}\right) \tag{25}$$

$$g^{[L]'}(Z^{[l]}) = \frac{d}{dZ^{[l]}}(ln(1 + e^{Z^{[l]}})) = \frac{1}{1+e^{-Z^{[l]}}} \tag{26}$$

3.6.8. Derivative of the Activation Functions

3.6.8.1. Derivative of the Swish Function

The function *swishDerivative()* implements Equation (27), which is the derivative of the Swish function. The function *swishDerivative()* takes in one argument in the form of a matrix or a vector and returns $\frac{dA^{[L]}}{dZ^{[L]}}$ as a vector in the second part of the neural network model.

$$\frac{df(x)}{dx} = \frac{d}{dx}\left(\frac{x}{1+e^{-0.5X}}\right) = \left(\frac{1+e^{-0.5X}+0.5\ e^{-0.5X}x}{(1-e^{-0.5X})^2}\right) \tag{27}$$

3.6.8.2. Derivative of the Softplus Function

The derivative of the softplus function given by Equation (28) is implemented by the function *sigmoid()* in the programm. The function *sigmoid()* takes in one argument in the form of a matrix or a vector and returns $\frac{dA^{[l]}}{dZ^{[l]}}$ as a matrix or vector in the hidden layers of the second part of the neural network model.

$$\frac{df(x)}{dx} = \frac{d}{dx}(ln(1 + e^x) = \frac{e^X}{1 + e^X} = \frac{1}{1 + e^{-x}} \tag{28}$$

3.6.8.3. Derivative of the ReLu Function

The derivative of ReLU function by Equation (29) is implemented through the function *reluDerivative(),* which takes in one argument in the form of a matrix or a vector to return $\frac{dA^{[L]}}{dZ^{[L]}}$ as a vector in the first part of the neural network model.

$$\frac{d}{dx}(max(1, x)) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{29}$$

3.6.8.4. Derivative of the leaky ReLu Function

The derivative of leaky ReLU function given by Equation (30) is implemented by the function *leakyReluDerivative()* in the programm. The function *leakyReluDerivative() takes* in one argument in the form of a matrix or a vector and returns $\frac{dA^{[L]}}{dZ^{[L]}}$ as a matrix or vector in the hidden layers of the first part of the neural network model.

$$\frac{d}{dx}(max(0.01x, x)) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0.01 & \text{if } x < 0 \end{cases} \tag{30}$$

3.6.9. Backward Propagation

The backpropagation phase during the training of the neural network is implemented through the function *backProp().* The function *backProp()* takes in five arguments. The first argument is the vector of actual values of COVID-19 contagion of shape (output size, number of examples), the second argument is the dictionary of activation values A and linearization values Z. The third argument is the input numpy array of shape (input size, number of examples). The fourth argument is the dictionary of weights and biases. The fifth argument is the string indicating the name of the activation function in the output layer, hence indicating which part of the neural network to perform based on the name of the output layer activation function. The function *backProp()* returns a dictionary of derivatives of the matrix Z -dZ-, the matrix of weights W -dW-, and the vector of biases B -dB. In fact, the function *backProp()* holds the code that implements the mathematical Equation (16), (17), (18), (19), (20), (21), (22), (23), (24), (25), (26), (27), (28), (29) and (30).

3.6.10. Parameter Update

The *function parameter()* fulfills the role of updating the parameters -weights and biases- of the neural network based on the result of the gradient descent yielded by the backpropagation. The *function parameter()* implements Equation (31) and (32).

$$W^{[l]} = W^{[l]} - (lr(dW^{[l]})) \tag{31}$$

$$b^{[l]} = b^{[l]} - (lr(db^{[l]})) \tag{32}$$

Where, $[l]$ denotes the layer l in the neural network, lr denotes the learning rate or the step size, $W^{[l]}$ is the weight matrix in layer l, $b^{[l]}$ is the bias vector in layer l

3.6.11. All in the Loop

The function *nnModelTrain()* uses the technique of iterative loop to implement the forward propagation, the cost function, the backpropagation, the parameter update and prediction at each iteration.

The function *nnModelTrain()* takes in six arguments. The first argument is a pandas dataframe from which to retrieve feature and target data points. The second argument is the name of the column from which to retrieve target data.The third argument is the dictionary containing initial parameters: weights and bias. The fourth argument is the activation function to determine the part of the neural network to implement. The fifth argument is the number of iterations. And the sixth argument is the learning rate.

The function *nnModelTrain()* returns five entities, whether collective or singular. The first entity is the dictionary of parameters learned by the neural network. The second entity is the dictionary of linearization and activations calculated by the neural network during the last iteration. The third entity is the dictionary of gradients learned by the neural network. The fourth entity is the list containing the number of iterations. And the fifth entity is the list containing costs per iteration.

3.6.12. The Coefficient of Determination

The coefficient of determination or R-squared score helps to measure the goodness of the fit of a model. The coefficient of determination uses the sum of squared regression (SSM) and the total sum of squares (TSS). Theen, the quotient of the sum of squared regression (SSM) and the total sum of squares (TSS) is subtracted from the unitary cardinal number. The closer the difference is to that unitary cardinal number, the betwer is the fitting regression. The R-squared score is calculated based on Equation (33). That Equation (33) is developed into a Python's code in order to calculate the coefficient of determination.

$$R^2 = 1 - \frac{SSR}{TSS} = 1 - \frac{\sum_{i=1}^{m} (y^{(i)} - \hat{y}^{(i)})}{\sum_{i=1}^{m} (y^{(i)} - \bar{y})} \quad (33)$$

Where:

- R^2 is the coefficient of determination or R-squared score
- SSR is the sum of squared regression
- TSS is the total sum of squares
- i is is the i^{th} training example
- m is the number of training examples
- $y^{(i)}$ is the actual contagion intensity value for i^{th} the training example
- $\hat{y}^{(i)}$ is the predicted contagion intensity value for the i^{th} training example
- $\bar{y}$ is the average of actual contagion intensity values

4. RESULTS

The final pandas dataframe "datum1", upon which the statistical analysis, the visual analysis and the training of the neural network are performed is made of 58 rows and 85 columns. The zip code area of 60666 of O'Hare International Airport is dropped. And among the 85 columns, 84 of them are the columns used for the prediction and one column "contagion intensity" is used as the target variable.

The target variable has a mean of 52.654579, standard deviation of 53.086, minimum of 0.609, maximum of 237.190, first quantile of 14.20, median of 39.459, third quantile of 73.402, mode of 0.609, mean absolute deviation of 37.008,

variance of 2818.160, kurtosis of 3.997, skewness of 1.935 and covariance of 2818.160355.

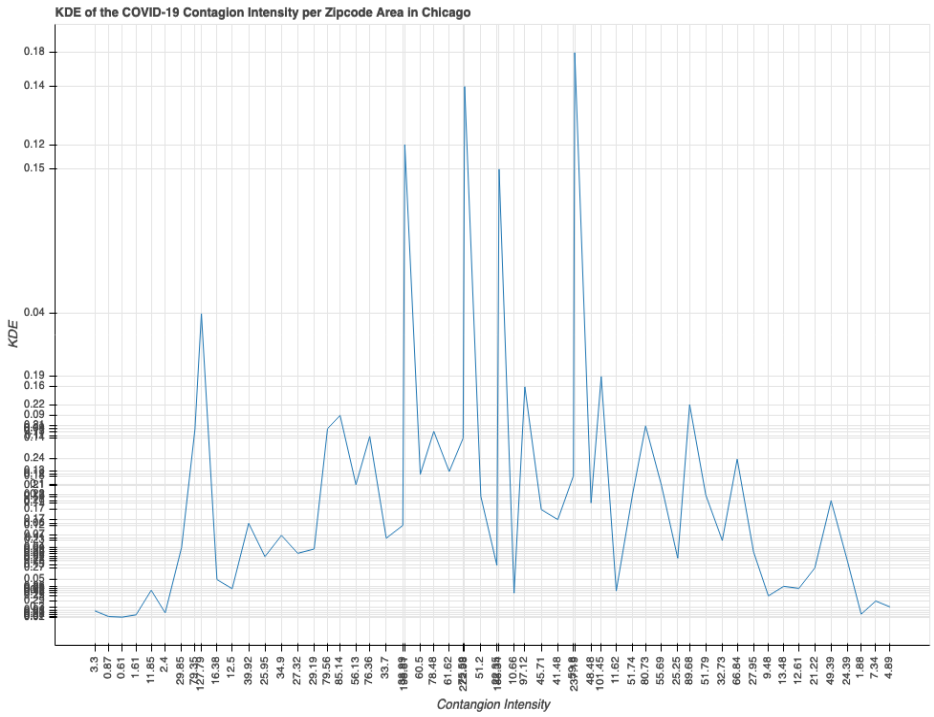


Figure 1. *The KDE plot of the target variable: COVID-19 Contagion intensity per Chicago's zip code area .*

Overall, the data points composing the target variable have a general average of kernel density estimation of 0.149.

The mean of all correlations between the feature variables and the target variable is 0.126.

By grouping the feature columns according to a number of characteristic criteria, the correlation between the COVID-19 contagion intensity and the variables of the property of the infrastructure investment is -0.0229 . That correlation between the government response parameters with the target variable is 0.0441. Variables distinguishing capital investment have a correlation average of -0.0822 with the

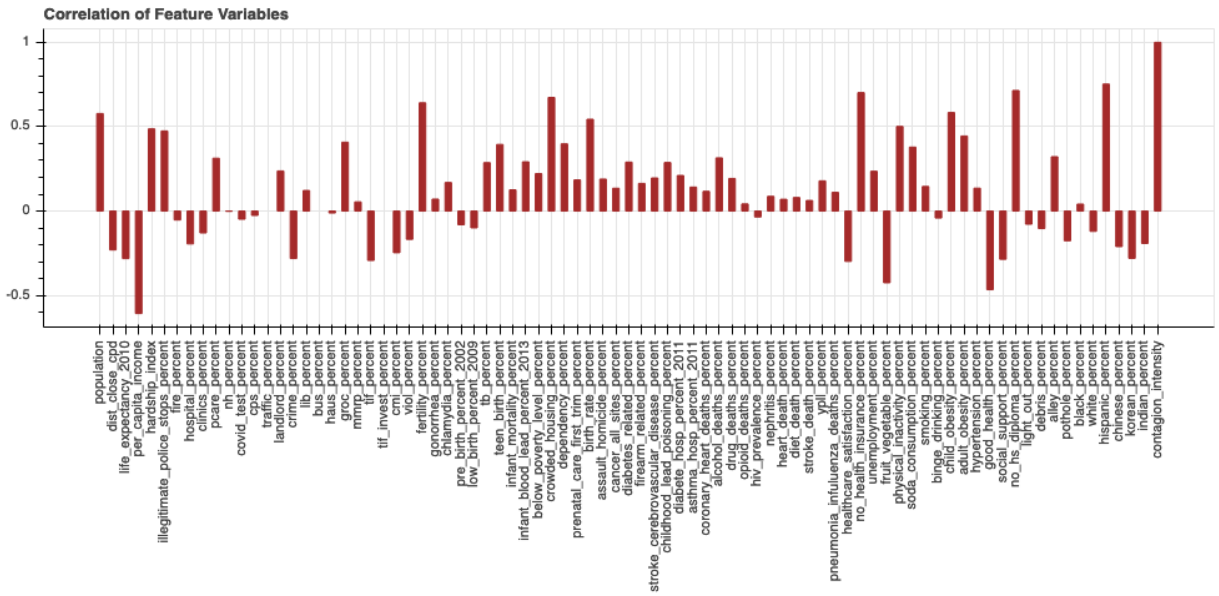


Figure 2. *The plot of correlation values between the target variable (contagion intensity per Chicago's zip code area) and the feature variables.*

COVID-19 contagion intensity. The parameter of safety scores the average of correlation of 0.0272 with the target variable. The statistical analysis of clinical care shows that its characteristic columns, with the COVID-19 contagion intensity, have an average correlation of 0.198. The elements of mortality show that its

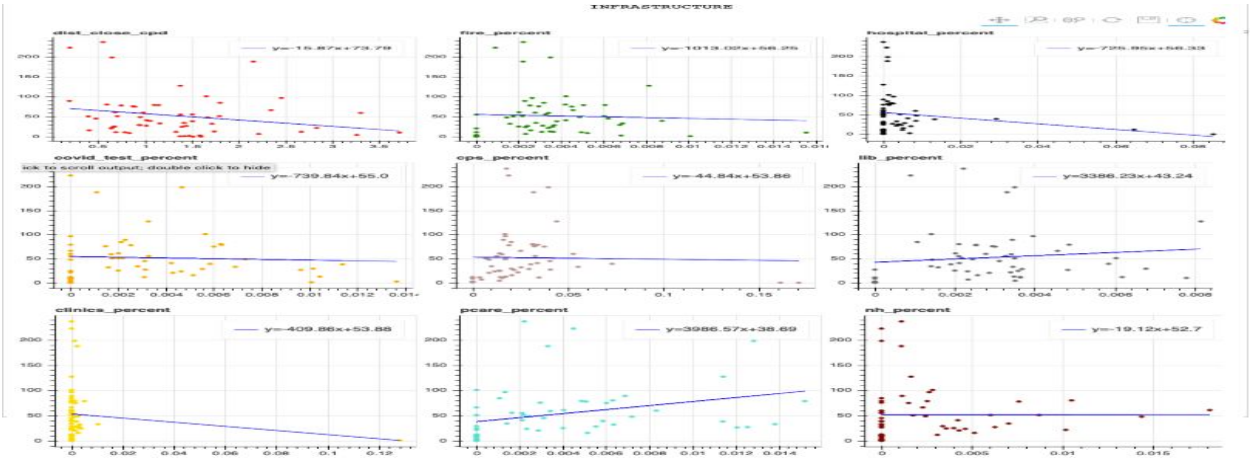


Figure 3.a. *The plot between the target variable (contagion intensity per Chicago's zip code area) and the feature variables characteristic of infrastructure.*

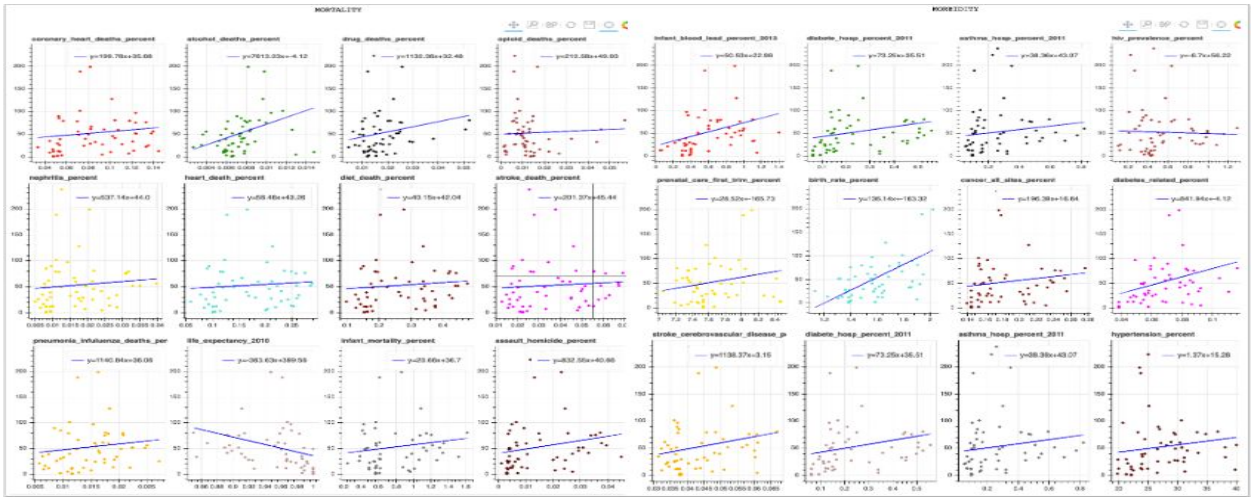


Figure 3.b. *The plot between the target variable (contagion intensity per Chicago's zip code area) and the feature variables characteristic of government response.*

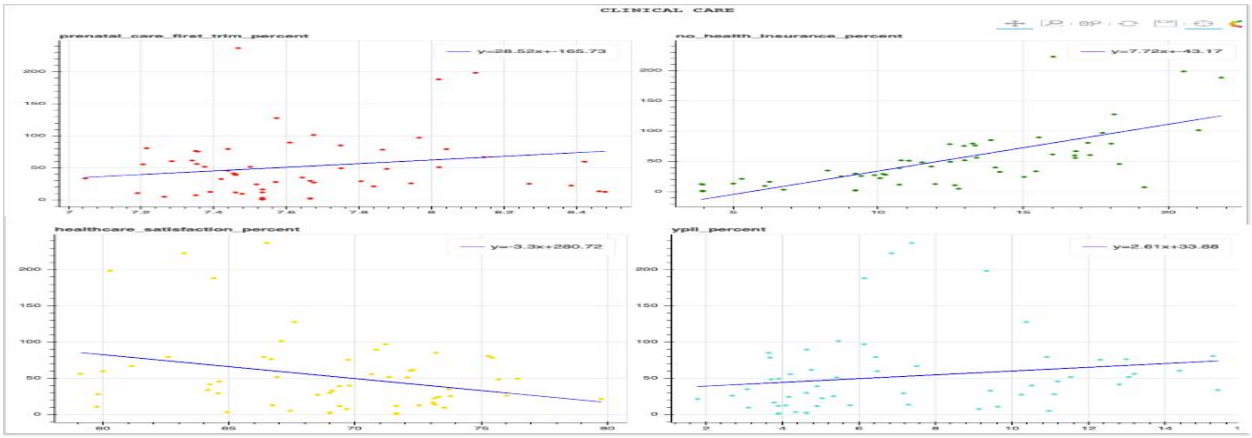


Figure 3.c. *The plot between the target variable (contagion intensity per Chicago's zip code area) and the feature variables characteristic of clinical care.*

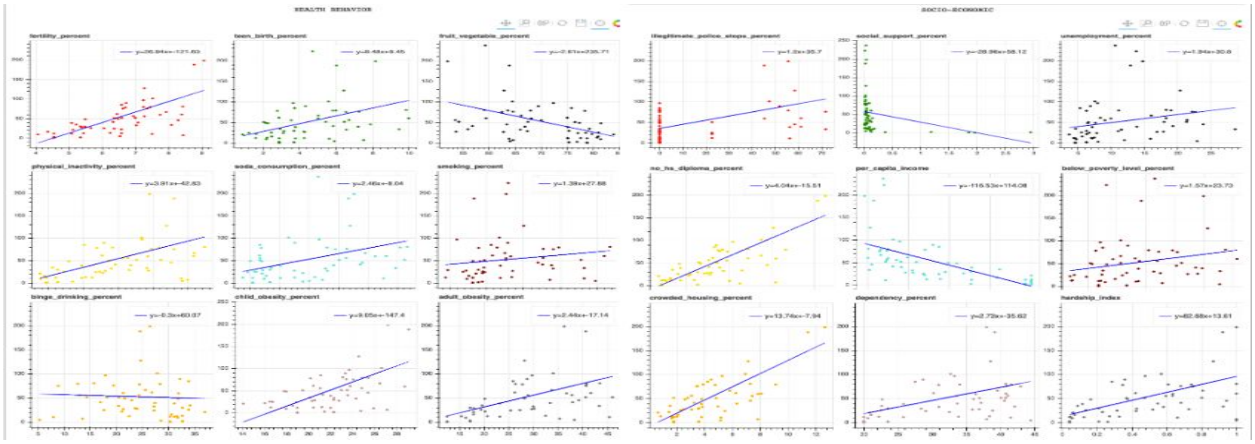


Figure 3.d. *The plot between the target variable (contagion intensity per Chicago's zip code area) and the feature variables characteristic of mortality and morbidity.*

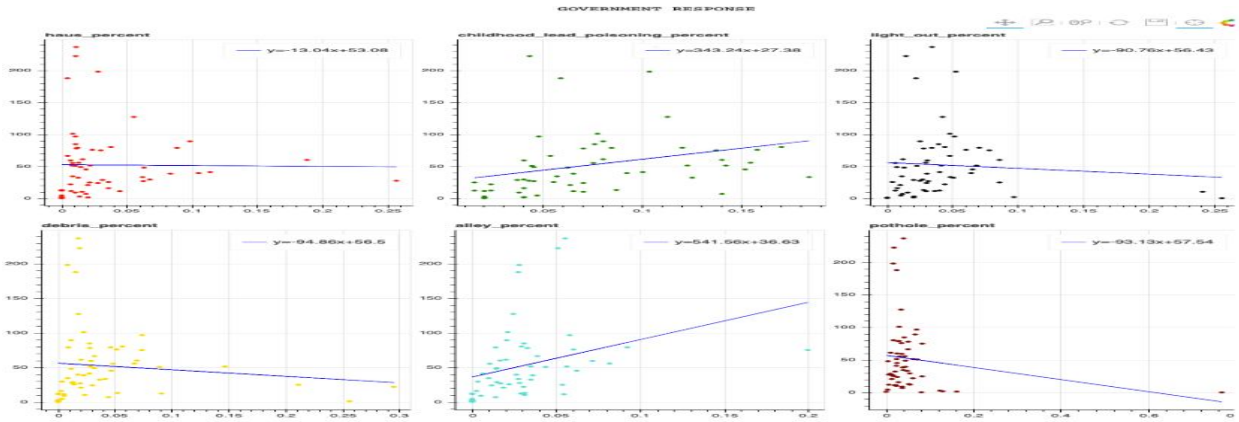


Figure 3.f. *The plot between the target variable (contagion intensity per Chicago's zip code area) and the feature variables characteristic of health behavior and socio-economic aspects.*

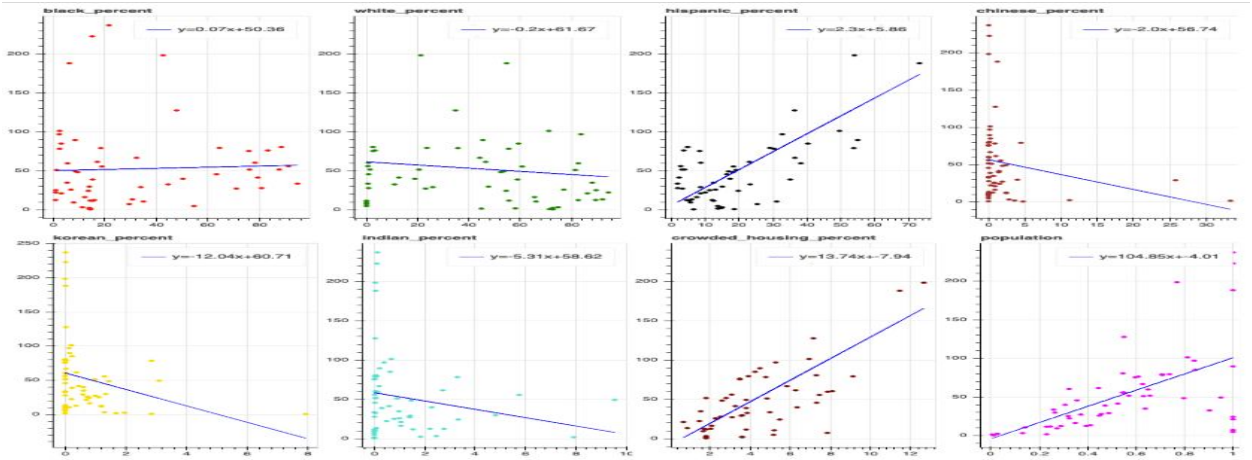


Figure 3.f. *The plot between the target variable (contagion intensity per Chicago's zip code area) and the feature variables characteristic of demography.*

correlation of 0.102. The morbidity ensemble indicates that its characteristic columns, with the COVID-19 contagion intensity, have an average correlation of 0.220.

Health behavior-a subcategory of health indicators in this study- shows that the average of all correlations of its properties with the COVID-19 contagion intensity have an average correlation of 0.298.

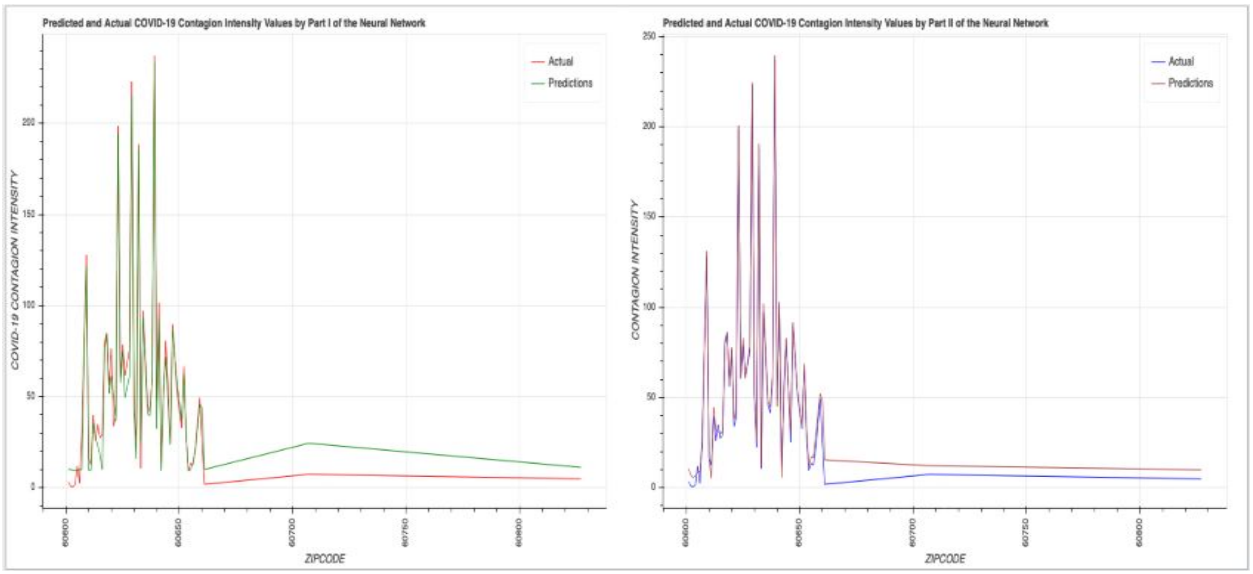


Figure 4. *The plot of the predictions against the actual values of the target variable: COVID-19 contagion intensity per Chicago's zip code area.*

The set of parameters composing the socio-economic aspect indicates the average correlation of 0.273, while those of demography score the average correlation of 0.181 with the target variable. This category of socio-economic aspects features measurements relative to the society, mores, routines and economic activities.

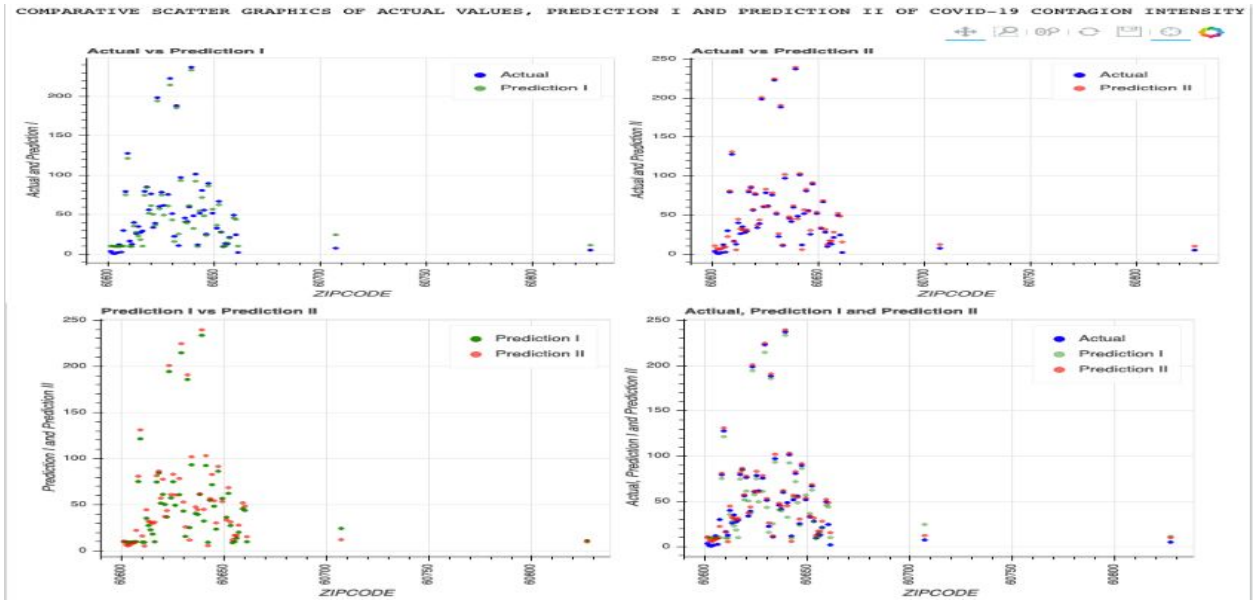


Figure 5. *Comparative Scatter plots of actual values, prediction I and prediction II of COVID-19 contagion intensity per Chicago's zip area.*

The application of machine learning's technique of the two-part neural network yields predictions that post a mean squared error of 37.443 by the first part of the neural network model. The second part of the neural network model yields a mean squared error of 9.747

The coefficient of determination or R-Squared score from the first part of the neural network is 0.974. The second part of the neural network posts an R-Squared score of 0.990.

The measure of the Correlation between target variable and prediction of the first part of the neural network is 0.988, whereas 0.996 is the correlation between target variable and prediction of the second part of the neural network.

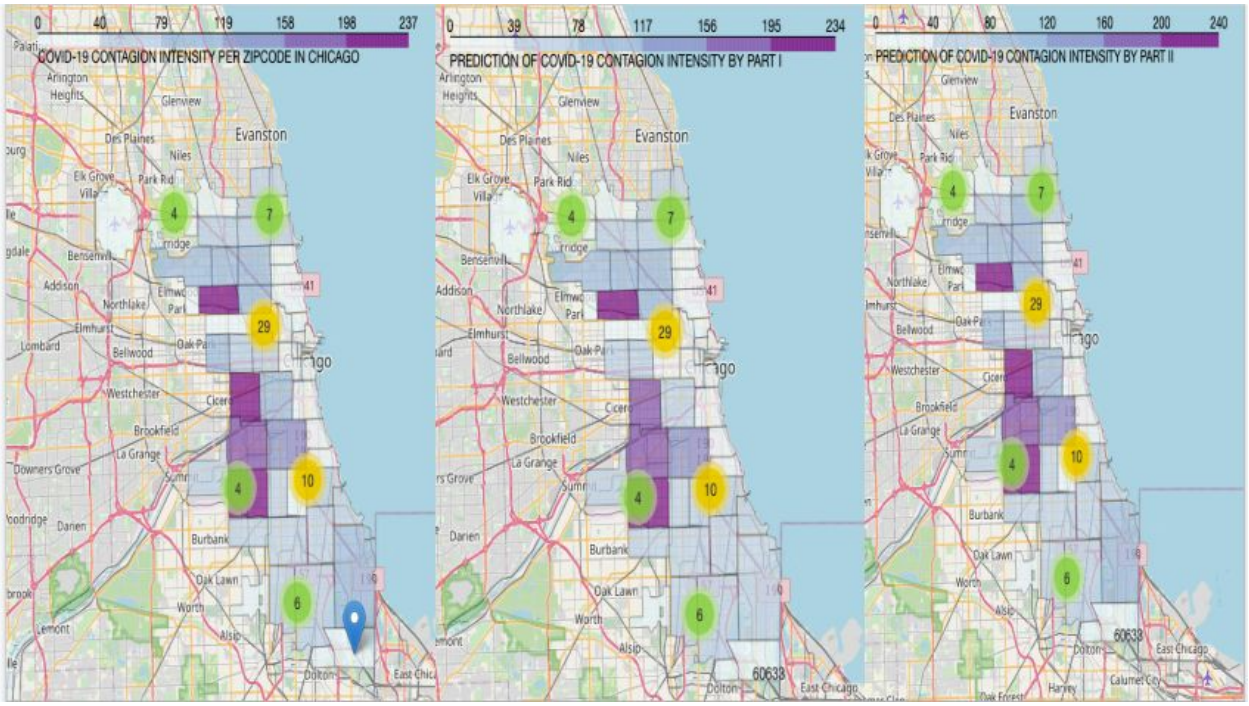


Figure 6. *The color map indicates the comparison of color projection based on the values of f COVID-19 contagion intensity per Chicago's zip area. On the left, it is the actual values. In the middle, it is the prediction by the first part of the neural network. On the right, it is the prediction by the second part of the neural network.*

The leaky ReLU function is utilized as the activation function for the hidden layers in the first part of the neural network model. In that part, ReLU function performs as the activation function for the output layer. In the first part of the neural network, the problem of local minimum is encountered, where the value of the mean squared error remains constant. With the application of ReLU and leaky ReLU in the first part of the neural network, it takes approximately 4,600 iterations to overcome the local minimum where the value of the mean squared error remained at 1409.12.

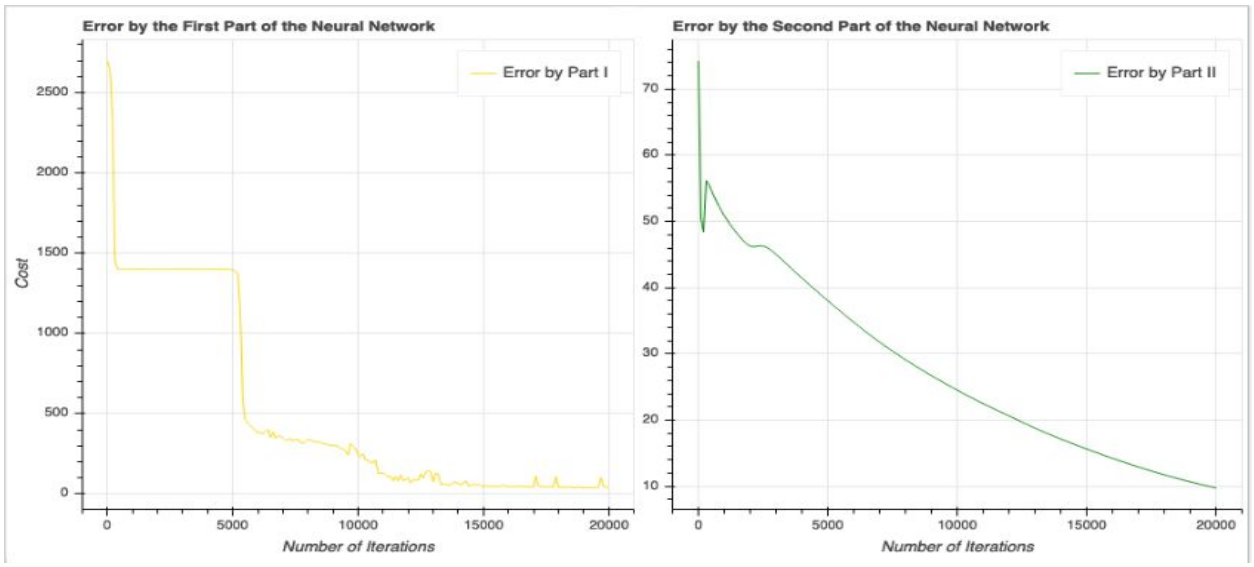


Figure 7. *The mean squared error by each part of the neural network model per iteration. On the left, it is by the first part, while on the right, it is by the second part.*

In the second part of the machine learning model, it is where the function of softplus is featured as the activation function for hidden layers in that second part of the neural network model. In that second part of the neural network model, the swish function works as the activation function for the output layer, which is ultimately the final output layer. In the second part of the neural network, the problem of vanishing gradient descent is encountered at iteration 200, where the value of the minimum squared error increases from 49.2 to 56.07.

5. DISCUSSION

5.1. Observation

The number of training examples is limited to 58, which is therefore the number of zip codes areas in Chicago. Given the fact of that limited number of training examples, it is impracticable to divide the training examples into cross training dataset, cross validation dataset and test dataset. Consequently, this could lead to the problem of overfitting.

Another aspect to take into consideration is that most of the data available is per community area. As one community area could be split into several zip code areas, it is compulsory to utilize averages of data from community areas when assigning data points to zip code areas. This could also lead to the compromise of data integrity as there is no available data on the percentage of a given community area population leaving in each zip code area it is divided into.

The visual insight of the map of Chicago about the COVID-19 contagion intensity reveals that the most affected zipcodes are neighboring next to each other. Those four neighboring and most affected zip code areas are 60609, 60623, 60629 and 60632. The fifth zip code area is 60639, which holds the maximum of COVID-19's contagion intensity as per date of July 26th, 2020.

The zip code area of 60629 holds the maximum of population, the maximum of COVID-19 positive cases total number and the maximum of the people tested positive of COVID-19 per day. The zip code area of 60623 is identified as the place in Chicago where there is the maxima of fertility rate percent, of crowded housing percent, of number of Chicago's Public Schools, of birth rate percent, of COVID's positive cases percent, of positive cases with regards of the number of tests, of the total number of deaths due to COVID-19, of COVID-19 related deaths per day, of

people without a high school diploma per 100 residents. Yet, the zip code area of 60623 is where there is the minimum of per capita income, of the hardship index and of the percentage of people who have easy access to fruits and vegetables. The zipcode of 60639 is where there is a maximum number of people who underwent the COVID-19 test and the number of COVID-19's tests per day. The zip code area of 60632 is a locality where there is the maximum of people with no health insurance, the maximum of people with child obesity, and the maximum of the people identified as Hispanic or Latino.

There are other zip code areas that have statistics that are worthy to pay attention to. The zip code area of 60644 scores the maximum of the percentage of opioid-related deaths, of the percentage of drug-abuse-related deaths, of the percentage of low- weight births as per year of 2009, of the percentage of infant mortality, of the percentage of all_sites cancer, of the percentage of diabetes-related deaths, of the percentage of cerebrovascular strokes disease, of the percentage of people classified as being physically inactive, of the percentage of people with high soda consumption , of the percentage people who are smokers. By contrast, this zipcode area of 60644 is where the residents have the lowest life expectancy as per year of 2010, and the minimum of the percentage of people who are deemed to be in good health. Among those attention-deserving zip code areas, there is also the zip code area 60624 where there is the maximum of the percentage of teen births, the maximum of the percentage of chlamydia cases, the maximum of the percentage of people living below poverty level and the maximum of the percentage of people who were hospitalized due asthma complications as per year of 2011. The zip code area of 60621 is where there is the maximum of the percentage of assault and homicide cases, the maximum of the percentage of firearm-related cases, the maximum of the percentage of childhood lead poisoning, the maximum of the percentage of potential years lost due to death before the age of 75, the maximum of the percentage of people who are unemployed and the maximum of the percentage of people identified as African American or Black. The

zip code area of 60619 is where there is the maximum of the percentage of of nephritis cases, the maximum of the percentage of heart-related deaths, the maximum of the percentage of diet-related deaths, the maximum of the percentage of hypertension cases and the minimum of the percentage of people satisfied with the healthcare they receive. The zip code area of is the place for the maximum of the percentage of healthcare satisfaction, the maximum of the percentage of the easy access to fruits and vegetables, the maximum of the percentage of binge drinking, the minimum of the percentage of physical inactivity and the minimum of the percentage of people with no high school diploma. The zip code area of 60631 is the locality with the maximum of the percentage of people in good health conditions and the maximum of the percentage of people identified as Non_Hispanic White. It is important to bear in mind that the zipcodes of 60621, 60624 and 60644 are neighboring the zipcode of 60609, 60623, 60629, 60632 and 60639.

The dataset of the contagion intensity shows that, in general, the data points are 3368.47 far each number from the mean. To put much more precise meaning on that measurement, the mean absolute deviation (MAD) calculates the variation in the contagion intensity dataset as being 41.99. This MAD value demonstrates that the values in the dataset are highly spread out.

Since, the kurtosis of the target variable dataset is equal to 3.996, which is greater than 3, it would be fair to conclude that the distribution is mesokurtic. This value proves that the distribution is almost similar to that of the normal distribution. This trend means that the extreme values of the distribution are characteristically almost similar to that of a normal distribution.

With the skewness of the target variable equal to 1.935 - a value greater than 1-, it proves that the data of the target variable are highly and positively skewed. This distribution indicates that the right tail of the data distribution is heavier than the left tail.

The average of the covariance between the feature variables and the target variable is 64.215. This positive value indicates that in general there is a positive relation , and the variability of the two variables goes in the same direction.

The average value of the Pearson Correlation Coefficient equals 0.126. This value quantifies the degree of the relationship between the feature variables and the target variable to be of positive correlation in general. The majority of the feature variables have a Pearson Correlation Coefficient of less than 0.6 with the target variable.

The set of infrastructure investment feature variables is composed of Chicago Police Department, fire stations, hospitals, clinics, primary care dispensaries, COVID-19 test sites, public schools, nursing homes, and libraries. Except for the primary care facilities, a quick look at the graphics realises that there is a negative correlation between the COVID-19 contagion intensity and the other parameters of infrastructure investment. The average of the correlation between the contagion intensity and the variables characteristic of the infrastructure investment is -0.0229. This translates that the higher the infrastructure investment, the lesser the spread of the novel COVID-19.

The set of government response feature variables comprises the measurement of aspects that require the government intervention for the sustainability of the wellbeing of the residents. This set comprises affordable rental housing, childhood poisoning level, complaints about non-functioning street lights, complaints about non-functioning alley lights, presence of tree debris and potholes in streets. This set of parameters indicating the level of the government response sends a mixed message. For example, in the case of complaints, it indicates that one interval of number of complaints percent in a given zip code spans across all the ranges of the COVID-19 contagion intensity. However, in the case of childhood lead poisoning, it sends a clear message: the higher the level of childhood lead poisoning in a given zip code area, the higher the level of COVID-19 contagion intensity. The result of this

heterogeneous aspect is that the average correlation between the government response parameters with the target variable is 0.0441.

Capital investment set contains data about microfinance lending, tax increment financing, micro-market recovery program and groceries. The domain of business per capita and microfinance lending per capita has the majority of data falling into the interval of 0 and 0.005, hence making it difficult to draw an accurate observation about its correlation with the COVID-19 contagion intensity. At the other hand, the number of groceries per 100 residents indicates that the higher the number of groceries in a given zip code, the higher the COVID-19 contagion intensity. However, the tax increment financing reverses the course. Understandably, the higher the tax increment financing , the lesser the COVID-19 contagion intensity in a given zip code. Thus, those mixed trends result in a correlation average of -0.043.

The collection of safety features englobes traffic accidents per 100 residents, problematic landlords per 100 residents, crimes per 100 residents, violations per 100 residents, assault and homicide per 100 residents, and firearm related crimes per 100 residents. In this collection, the numbers yielded by crime and violation per 100 residents are inconclusive. On the other hand, the majority of data is concentrated between 0.00 and 0.05 and that majority corresponds to the COVID-19 contagion intensity ranging between 0.00 and 300. Furthermore, the higher the number of traffic accidents, assault and homicide, and firearm crimes corresponds to the increase of the COVID-19 contagion intensity, as well. The end product is that the result of this inconsistency yields an average of correlations of 0.0285.

The visual analysis of clinical care - a subcategory of the health sector- reveals two trends. The first trend is that when the percentage of people satisfied with the health care increases, the contagion intensity decreases. The second trend reveals that the higher COVID-19 contagion intensity, the higher the percentage of people

who are losing potential years due to premature deaths, people who need prenatal care during the first trimester of pregnancy and people who have no health insurance. The statistical analysis of clinical care shows that its characteristic columns, with the COVID-19 contagion intensity, have an average correlation of 0.198, which translates to the fact that the higher the lack of appropriate clinical care, the higher the COVID-19 intensity of contagion in a given Chicago's zip code area.

The linear fitting of mortality - another subcategory of health sector in this study- versus the COVID-19 contagion intensity reveals that the percentage of coronary heart deaths, infant mortality, pneumonia influenza deaths, alcohol deaths, drug deaths, opioid deaths, nephritis, heart death, diet death, assault homicide and stroke death increases as the COVID-19 contagion intensity increases. However, the slope of the linear regression of the life expectancy versus the COVID-19 contagion intensity has a negative slope. Hence, indicating that the higher the life expectancy, the lesser the contagion intensity. In general, the statistical analysis of mortality shows that its characteristic columns, with the COVID-19 contagion intensity, have an average correlation of 0.102, which, so far, means that the higher the causes of mortality in a given Chicago's zip code area, the higher the contagion intensity of COVID-19.

The statistical analysis of morbidity indicates that its characteristic columns, with the COVID-19 contagion intensity, have an average correlation of 0.220. This positive trend in the correlation of morbidity - the third subcategory of the health sector in this study- informs that every characteristic property of morbidity in this study increases as the COVID-19 contagion intensity gets higher.

The statistical analysis of health behavior-the last subcategory of health indicators in this study- shows that the average of all correlations of its properties with the COVID-19 contagion intensity have an average correlation of 0.298. As per observation, it indicates that the contagion intensity increases as the percentage of

fertility, teenage births, physical inactivity, soda consumption, smoking, binge drinking, childhood obesity, and adult obesity increases. However, the observation indicates that the COVID-19 contagion intensity reduces as more people have easy access to fruits and vegetables.

The set of socio-economic features is made of 9 elements, namely; the percentage of illegitimate police stops, the percentage of people on social support, the percentage of unemployed people, the percentage of of people without high school diploma, per capita income, the percentage of people living below poverty level, the number crowded housing per 100 residents, the percentage of people in dependency, and hardship_index. The correlation between the socio-economic features and the COVID-19 contagion intensity corresponds to the observation fitting with the rationale going like "the better the socio-economic conditions, the lesser the intensity of COVID-19 contagion". The average correlation of 0.273 comes as a result of two of the features - per capita income and life expectancy- with a negative correlation, while the rest depicts a positive correlation.

This set of demography is an ensemble composed of the race-ethnicity designation of peoples groups, namely; African-American or Black people percentage per zip code, Non-Hispanic White people percentage per zip code, Hispanic or Latino people percentage per zip code, Chinese people percentage per zip code, Korean people percentage per zip code, and Asian Indian people percentage per zip code. In addition to those ethnic designation, this set also comprises the number of population per zip code and the number of crowded housing.

The average correlation of 0.181 comes as a result of mixed results. The positive correlation is scored by the criterion of percentage of African-American or Black population per zip code, percentage of Hispanic or Latino population per zip code, percentage of crowded housing and population per zip code. The negative correlation comes from the criteria of Non-Hispanic White people percentage per

zip code, Chinese people percentage per zip code, Korean people percentage per zip code, and Asian Indian people percentage per zip code.

The neural network model composed of two parts yields positive results. Both parts of the neural network train by making 20,000 iterations each. The first part of the neural network has the activation functions of ReLu and leaky ReLU. Both parts of the neural network predict the maximum of the COVID-19 contagion intensity (233.550 and 239.507 respectively) to be in the zipcode area of 60639, which is the same as of the actual value (237.190). However the first part predicts the minimum of COVID-19 contagion intensity (9.125) to be in the zipcode area of 60654, where the second part of the neural network predicts the minimum of COVID-19 contagion intensity (5.374) to be in the zipcode area of 60611. The actual minimum of contagion intensity is 0.609 in the zipcode area of 60603 as of July 26, 2020.

The correlation between the target variable and prediction of the first part of the neural network is 0.988. The target variable and prediction of the second part of the neural network give a correlation of 0.997.

If the mean of the actual values of COVID-19 contagion intensity is 52.655, the first part of the neural network yields predictions that have a mean of 50.687 and the second part of the neural network yields the mean of 55.367. Basically, the mean of the of actual values of the COVID-19 contagion intensity is 2.068 higher than the mean of COVID-19 contagion intensity predicted by the first part of the neural network model, whereas it is 2.712 lesser than the predictions by the second part of the neural network model.

In terms of accuracy, the coefficient of determination by the first part is 0.974, and 0.990 is the r-squared error by the second part of the neural network. The close to 1.00 the coefficient of determination gets, the better is the training model. The value of coefficient of determination yielded by the second part is closer to one.

As of the mean squared error, the first part of the neural network encounters two challenges. The first challenge is encountering a local minimum that takes more 5,000 iterations to overcome. However, the first part does a robust job where the mean squared error goes down from 2771.038 to 36.492. The second challenge consists of a bumpy training path where the training error augments before going back down to a lesser value on several occasions. The second part of the neural network, which performs through the activation functions of Swish and Softplus, has a much smoother training path where the training error keeps going down. At the end, the mean squared error goes down from 74.250 to 9.747.

5.1. Recommendation

Given the fact that the zip code areas of 60609, 60623, 60629, 60632 and 60639 have the highest COVID-19 contagion intensities, it would be appropriate to set up more COVID-19 testing sites in those areas. The zip code areas of 60619, 60621, 60624 and 60644, which also have alarming scores in a number of aspects and neighboring those aforementioned zip code areas, also need more COVID-19 testing sites. For a much better accurate prediction, there is a need to have access to up-to-date data. That issue of data augmentation would also include data on other critical aspects of life. Thereso, it could help to work on the matter of overfitting with more features in the training set.

It is also noteworthy to collect data on mobility between Chicago zip code areas and other parts. This could help in having a more accurate SIR model.

Algorithmically, it may help to consider the hybrid of the functions of swish, softplus, ReLU and leaky ReLU when envisaging the deployment of deep neural networks to predict continuous values.

6. Conclusion

The application of a two-part neural network model featuring the hybrid of four activation functions -ReLU and leaky ReLU for the first part, Swish and Softplus for the second part- provides a machine learning model of deep neural network that is efficient in predicting continuous values and able to overcome the issue of vanishing gradient descent and local minima or maxima jamming. That efficiency performs better with rescaling features with data points that have values that are higher than the maximum value of the target variable.

With this project only limited on the level of zip code areas of the City of Chicago, the identical technique could extrapolated on a much larger scale, such as the national level to make predictions of contagion intensity -not only of coronavirus, but also of any other contagious disease- based on properties of preexisting data on socio-economic aspects, health indicators, administrative policies and physical environment, rather than being only limited on real-time data on pathogenic properties particular to a given pandemic disease.

References

1. Chicago, City of. “COVID-19 Cases, Tests, and Deaths by ZIP Code: City of Chicago: Data Portal.” Chicago Data Portal, July 24, 2020. https://data.cityofchicago.org/Health-Human-Services/COVID-19-Cases-Tests-and-Deaths-by-ZIP-Code/yhhz-zm2v.
2. “Real Estate For Sale by Chicago Zip Code.” var-site-name. Accessed July 31, 2020. https://www.seechicagorealestate.com/chicago-zip-codes-by-neighborhood.php.
3. Chicago, City of. “Fire Stations: City of Chicago: Data Portal.” Chicago Data Portal, August 21, 2011. https://data.cityofchicago.org/Public-Safety/Fire-Stations/28km-gtjn.
4. City of Chicago. “Chicago Department of Public Health Clinic Locations: City of Chicago: Data Portal.” Chicago Data Portal, August 3, 2017. https://data.cityofchicago.org/Health-Human-Services/Chicago-Department-of-Public-Health-Clinic-Locatio/kcki-hnch.
5. “Public Health Services- Chicago Primary Care Community Health Centers: City of Chicago: Data Portal.” Chicago Data Portal, April 22, 2014. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Services-Chicago-Primary-Care-Commun/cjg8-dbka.
6. “Medicare/Medicaid Nursing Homes & Rehab in Cook County (Page 1).” The Big List of Nursing Homes in Cook County, IL. Accessed July 31, 2020. https://www.dibbern.com/nursing-homes/illinois/cook_county_illinois_nursing_homes.htm.
7. Chicago, City of. “COVID-19 Testing Sites: City of Chicago: Data Portal.” Chicago Data Portal, July 13, 2020. https://data.cityofchicago.org/Health-Human-Services/COVID-19-Testing-Sites/thdn-3grx.
8. Schools, Chicago Public. “Chicago Public Schools - School Locations SY1920: City of Chicago: Data Portal.” Chicago Data Portal, August 14, 2019. https://data.cityofchicago.org/Education/Chicago-Public-Schools-School-Locations-SY1920/tz49-n8ze.
9. Chicago, City of. “Traffic Crashes - People: City of Chicago: Data Portal.” Chicago Data Portal, July 30, 2020. https://data.cityofchicago.org/Transportation/Traffic-Crashes-People/u6pd-qa9d.
10. City of Chicago. “Problem Landlord List: City of Chicago: Data Portal.” Chicago Data Portal, July 11, 2016. https://data.cityofchicago.org/Buildings/Problem-Landlord-List/n5zj-r44u.
11. Department, Chicago Police. “Crimes - 2001 to Present: City of Chicago: Data Portal.” Chicago Data Portal, July 30, 2020. https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2.

12. “Libraries - Locations, Hours and Contact Information - XML File.” Data.gov. Accessed July 31, 2020. https://catalog.data.gov/dataset/libraries-locations-hours-and-contact-information-f3c61/resource/c5a2baaa-8b3d-4155-9891-a54e0ac8b042.
13. Chicago, City of. “Business Licenses - Current Active: City of Chicago: Data Portal.” Chicago Data Portal, July 30, 2020. https://data.cityofchicago.org/Community-Economic-Development/Business-Licenses-Current-Active/uupf-x98q.
14. City of Chicago. “Affordable Rental Housing Developments: City of Chicago: Data Portal.” Chicago Data Portal, July 23, 2020. https://data.cityofchicago.org/Community-Economic-Development/Affordable-Rental-Housing-Developments/s6ha-ppgi.
15. City of Chicago “Grocery Stores - 2013: City of Chicago: Data Portal.” Chicago Data Portal, August 26, 2013. https://data.cityofchicago.org/Community-Economic-Development/Grocery-Stores-2013/53t8-wyrc.
16. City of Chicago. “Micro-Market Recovery Program - Addresses: City of Chicago: Data Portal.” Chicago Data Portal, July 30, 2020. https://data.cityofchicago.org/Community-Economic-Development/Micro-Market-Recovery-Program-Addresses/cf2f-mmzv.
17. “Tax Increment Financing (TIF) Funded RDA and IGA Projects - Dashboard: City of Chicago: Data Portal.” Chicago Data Portal. Accessed July 31, 2020. https://data.cityofchicago.org/Community-Economic-Development/Tax-Increment-Financing-TIF-Funded-RDA-and-IGA-Pro/urnb-brza.
18. Chicago, City of. “Ordinance Violations: City of Chicago: Data Portal.” Chicago Data Portal, July 30, 2020. https://data.cityofchicago.org/Administration-Finance/Ordinance-Violations/6br9-quuz.
19. Health, Illinois Department of Public. “Public Health Statistics - General Fertility Rates in Chicago, by Year, 1999-2009: City of Chicago: Data Portal.” Chicago Data Portal, October 5, 2012. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-General-fertility-rates-i/g5zk-9ycw.
20. “Public Health Statistics- Gonorrhea Cases for Females Aged 15 - 44 in Chicago, by Year, 2000 - 2014: City of Chicago: Data Portal.” Chicago Data Portal, January 8, 2016. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Gonorrhea-cases-for-femal/cgjw-mn43.

21. “Public Health Statistics- Gonorrhea Cases for Males Aged 15-44 in Chicago, by Year, 2000 - 2014: City of Chicago: Data Portal.” Chicago Data Portal, January 8, 2016. https://data.cityofchicago.org/Health-Human-Services/Public-health-statistics-Gonorrhea-cases-for-males/m5qn-gmjx.
22. “Public Health Statistics- Chlamydia Cases among Females Aged 15-44 in Chicago, by Year, 2000-2014.: City of Chicago: Data Portal.” Chicago Data Portal, January 8, 2016. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Chlamydia-cases-among-fem/bz6k-73ti.
23. Chicago, City of. “Public Health Statistics - Chlamydia Cases among Males Aged 15-44 in Chicago, by Year, 2000-2014: City of Chicago: Data Portal.” Chicago Data Portal, January 25, 2016. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Chlamydia-cases-among-mal/35yf-6dy3.
24. Illinois Department of Public Health (IDPH). “Public Health Statistics - Preterm Births in Chicago, by Year, 1999 – 2009: City of Chicago: Data Portal.” Chicago Data Portal, March 28, 2013. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Preterm-births-in-Chicago/rhy3-4x2f.
25. City of Chicago. “Public Health Statistics - Low Birth Weight in Chicago, by Year, 1999 – 2009: City of Chicago: Data Portal.” Chicago Data Portal, October 23, 2012. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Low-birth-weight-in-Chica/fbxr-9u99.
26. Vital statistics files produced by the Illinois Department of Public Health (IDPH). “Public Health Statistics- Life Expectancy By Community Area: City of Chicago: Data Portal.” Chicago Data Portal, June 16, 2014. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Life-Expectancy-By-Commun/qjr3-bm53.
27. Health, Public. “Public Health Statistics- Tuberculosis Cases and Average Annual Incidence Rate, Chicago, 2007- 2011: City of Chicago: Data Portal.” Chicago Data Portal, April 11, 2014. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Tuberculosis-cases-and-av/ndk3-zftj.
28. Illinois Department of Public Health (IDPH). “Public Health Statistics - Births to Mothers Aged 15-19 Years Old in Chicago, by Year, 1999-2009: City of Chicago: Data Portal.” Chicago Data Portal, March 28, 2013. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Births-to-mothers-aged-15/9kva-bt6k.

29. City of Chicago. “Public Health Statistics- Infant Mortality in Chicago, 2005– 2009: City of Chicago: Data Portal.” Chicago Data Portal, April 11, 2014. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Infant-mortality-in-Chica/bfhr-4ckq.
30. Childhood Lead Poisoning Prevention Program, Chicago Department of Public Health (CDPH). “Public Health Statistics - Screening for Elevated Blood Lead Levels in Children Aged 0-6 Years by Year, Chicago, 1999 - 2013: City of Chicago: Data Portal.” Chicago Data Portal, February 5, 2015. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Screening-for-elevated-bl/v2z5-jyrq.
31. Illinois Department of Public Health (IDPH) and U.S. Census Bureau. “Public Health Statistics-Selected Public Health Indicators by Chicago Community Area: City of Chicago: Data Portal.” Chicago Data Portal, May 30, 2013. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Selected-public-health-in/iqnk-2tcu.
32. Illinois Department of Public Health (IDPH). “Public Health Statistics- Diabetes Hospitalizations in Chicago, 2000 - 2011: City of Chicago: Data Portal.” Chicago Data Portal, August 6, 2012. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Diabetes-hospitalizations/vekt-28b5.
33. City of Chicago. “Public Health Statistics - Asthma Hospitalizations in Chicago, by Year, 2000 - 2011: City of Chicago: Data Portal.” Chicago Data Portal, September 17, 2012. https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Asthma-hospitalizations-i/vazh-t57q.
34. Bureau, U.S. Census. “Census Data - Selected Socioeconomic Indicators in Chicago, 2008 – 2012: City of Chicago: Data Portal.” Chicago Data Portal, September 12, 2014. https://data.cityofchicago.org/Health-Human-Services/Census-Data-Selected-socioeconomic-indicators-in-C/kn9c-c2s2.
35. Chicago, City of. “311 Service Requests - Street Lights - All Out - Historical: City of Chicago: Data Portal.” Chicago Data Portal, March 6, 2019. https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Street-Lights-All-Out-Histori/zuxi-7xem.
36. City of Chicago. “311 Service Requests - Tree Debris - Historical: City of Chicago: Data Portal.” Chicago Data Portal, March 6, 2019. https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Tree-Debris-Historical/mab8-y9h3.

37. City of Chicago. “311 Service Requests - Alley Lights Out - Historical: City of Chicago: Data Portal.” Chicago Data Portal, March 6, 2019. https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Alley-Lights-Out-Historical/t28b-ys7j.
38. City of Chicago. “311 Service Requests - Pot Holes Reported - Historical: City of Chicago: Data Portal.” Chicago Data Portal, March 5, 2019. https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Pot-Holes-Reported-Historical/7as2-ds3y.
39. Team, ZipAtlas.com Development. Percentage of Blacks (African Americans) in Chicago, IL by Zip Code. Accessed July 31, 2020. http://zipatlas.com/us/il/chicago/zip-code-comparison/percentage-black-population.htm.
40. Zip Atlas. Percentage of Whites in Chicago, IL by Zip Code. Accessed July 31, 2020. http://zipatlas.com/us/il/chicago/zip-code-comparison/percentage-white-population.htm).
41. Zip Atlas. Percentage of Hispanics in Chicago, IL by Zip Code. Accessed July 31, 2020. http://zipatlas.com/us/il/chicago/zip-code-comparison/percentage-hispanic-population.htm.
42. Chicago Health Atlas. Accessed July 31, 2020. https://www.chicagohealthatlas.org/indicators/hispanic-or-latino.
43. “Compartmental Models in Epidemiology.” Wikipedia. Wikimedia Foundation, July 28, 2020. https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology.