Forecasting of COVID-19 Data

Aruvi Puhazhendhi
puhazhendhi.a@northeastern.edu
Masters in Artificial Intelligence
Khoury College of Computer Sciences
Northeastern University, Boston

Abstract

World Health Organization has declared COVID-19 a pandemic on March 11th. As we are encountering ourselves in the middle of the biggest crisis of the last few decades, the governments are trying to forecast the impacts of the novel corona virus (COVID – 19). A crisis like this has not been observed after the last pandemic which was in 1918 and the Great Depression which was in 1929. Starting with medical researchers, businesses, government policy makers, environmentalists, everyone is trying to understand how the disease will propagate and how to mitigate the effects. Forecasts show that daily growth rates should be kept at least below 5 percent if we wish to see plateaus any time soon—unfortunately far from reality in most countries to date. To be able to predict the spread accurately, we need to model the spread of confirmed cases and at the same time also model the deaths and recoveries. Keeping in mind predictions can seldom be completely accurate, we can rely upon them to give an estimate of the direction with some level of confidence. In this paper we will use forecasting methods with an assumption that the data we have in hand is accurate, we will compare between different models and parameters to identify the ones that work best for the countries under consideration. We will be performing the analysis on 8 most affected countries and India. Additionally, we will also extract Reddit conversations on the topic to investigate if there any kind of racial biases.

Keywords

Covid-19 - Forecasting - Machine Learning - Auto-regression

Introduction

The effectiveness of the predictions largely depend on the amount of data we have on the subject. In our case of COVID-19, the data we have until this date is still very less compared to what is ideally required for a good prediction. Additionally, many countries and governments also tend to under report the number of affected cases and death so as to conceal the extent of the impact. However according to the reports, the number of infected cases and deaths are growing exponentially in most of the countries. In a few countries the healthcare facilities are already overburdened. In this time of urgency even a simplest model built on moderately reliable data can prove to be greatly useful for the governments to prepare accordingly and to motivate the wider public to adhere to the guidelines of social distancing basing on "scientifically" forecast-ed outcomes. In this paper, we therefore present simple auto regressive method to forecast the number of COVID-19 cases, under the assumption that data is legitimate and truthful. The goal is not to strive for meticulous accuracy nor to present our method as the state of the art, but simply to provide first insights and guidelines on elementary principles. We will be happy if our work motivates further research to yield more elaborate and accurate prediction methods. We will try out variations on auto regression models and compare the results to see which ones performs better for different countries.

Additionally, another part of the project will involve extracting all the Reddit threads on the topic "Coronavirus" and running a topic modeling analysis. The objective is to investigate major topics that are being discussed in the conversations and to identify racially discriminating conversations if there are any.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Background

The prediction algorithm that will be used is an auto-regressive model. It generally is used to describe certain time-varying processes in nature, economics, etc. The auto-regressive model specifies that the output variable depends linearly on its own previous values and on a stochastic term (an imperfectly predictable term); thus the model is in the form of a stochastic difference equation (or recurrence relation which should not be confused with differential equation). The way COVID-19 spreads, is from an individual to other people. The number of positive cases yesterday directly affects the number of positive cases today. Since the problem of COVID-19 depends on its own previous values to a great degree, and since this is a time-varying process an auto-regressive linear model is appropriate.

Related Work

There is a lot of attention on this topic and people are using various time series methods, RNN(recurring neural net-

works), SIR models and predictive analytics to arrive at a good prediction.

Traditional infectious disease prediction models mainly include differential equation prediction models and time series prediction models based on statistics and random processes. The currently widely studied and applied models include SI model, SIS model, SIR model and SEIR model, etc.(Jia et al.(2020)Jia, Li, Jiang, Guo, et al.)

Time series prediction models, based on statistics and random processes, predict infectious diseases by analyzing one-dimensional time series of infectious disease incidence, mainly including Auto-regressive Integrated Moving Average model (ARIMA), Exponential Smoothing method (ES), Grey Model (GM), Markov chain method (MC), etc. The widely used time series prediction model is ARIMA prediction model, which uses several differences to make it a stationary series, and then represent this sequence as a combination auto-regression about the sequence up to a certain point in the past.

Internet-based infectious disease prediction model Infectious disease surveillance research based on the Internet has begun to rise since the mid-1990s. It can provide information services for public health management institutions, medical workers and the public. After analyzing and processing, it can provide users with early warning and situational awareness information of infectious diseases(Brownstein et al.(2009)Brownstein, Freifeld, and Madoff).

Compared with the traditional prediction models, the Internet-based infectious disease prediction models have the advantages of real-time and fast, which can predict the incidence trend of infectious diseases as early as possible, and are suitable for data analysis of a large number of people. However, the sensitivity, spatial resolution and accuracy of its prediction need to be further improved. So Internet-based infectious disease prediction models cannot replace the traditional prediction models, and they can just be used as an extension of the traditional infectious disease prediction model(Milinovich et al.(2015)Milinovich, Magalhães, and Hu).

Project Description

Data

We are using the data that was uploaded by Kaggle. The data set consists of confirmed cases and fatalities due to COVID-19 across countries and provinces. The data is on a daily level and starts from last week of January 2020 and ends with data till few days ago (12th of April, 2020). However, we will be using only the data for only 9 countries. The evaluation data comes from John Hopkins CSSE data.

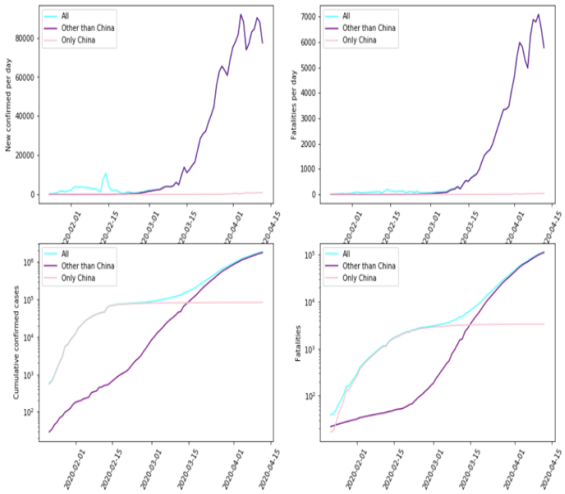


Fig0. A simple data exploration that shows the trend for all countries, only China and all countries other than China.

We are additionally using external data from sources like worldbank.org that consists of factors that might affect the transmission rate such as, starting dates of social lockdowns, population density numbers, average flight passengers, number of migrants etc.,

We used sklearn.'s TimeSeriesSplit to prepare our data into a time series format. We created two different data sets. One for confirmed cases and one for the number of deaths. Following this we forecast-ed the trends of confirmed cases and number of deaths separately.

Algorithm

Forecasting: We tried 2 algorithms on our data to forecast the Covid-19 confirmed cases and fatalities

1. Auto Regressive Elastic Net: Elastic-Net is a linear regression model trained with both 1 and 2-norm regularization of the coefficients. Elastic-net is useful when there are multiple features which are correlated with one another. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both. The choice between the weights are chosen using cross validation and hence the computational requirements are higher than both.

2. Random Forest Regression: A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the data set and uses averaging to improve the predictive accuracy and control over-fitting.

Evaluation

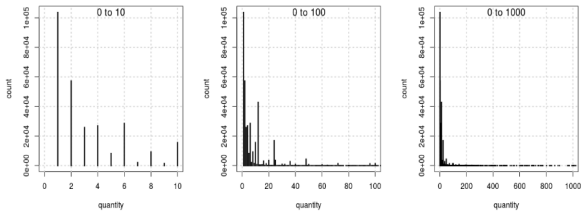
The evaluation metric we are using is root mean square log error (RMSLE). The RMSLE is calculated as

$$RMSLE(a,p) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2}$$

where n is the total number of observations, log(x) is the natural logarithm of x.

Adding 1 before taking the logarithm is a standard technique used to avoid evaluating log(0) which is undefined. Python has the convenience function log1p(x) = log(x+1) and its inverse expm1 = exp(x)-1.

RMSLE is commonly used in regression problems where the target variable distribution is highly skewed.



If we were to minimize the plain RMSE for these types of distributions, our learning algorithm would focus on minimizing the large but infrequent errors in the tails of the target distribution. The logarithm in the RMSLE helps to make the target variable distribution less skewed, thus allowing the learning algorithm to focus on the errors in a more common range of the distribution. RMSLE is the RMSE of the log1p-transformed target variable. This is an important property. It implies that minimizing RMSLE is the same as first applying the log1p transformation to our target variable, then minimizing the RMSE (which is a standard error metric).

Topic modeling: We extracted all the data under the subreddit "Coronavirus" using the Reddit API. We cleaned and pre-processed the data by removing stop words, tokenizing, lemmatizing, stemming etc. We have used two packages to build corpuses for topic modeling.

1. Bag of words: It is an algorithm that counts how many times a word appears in a document. It's a tally. Those word counts allow us to compare documents and gauge their similarities for applications like search, document classification and topic modeling. BoW is a also method for preparing text for input in a deep-learning net. BoW lists words paired with their word counts per document. In the table where the words and documents that effectively become vectors are stored, each row is a word, each column is a document, and each cell is a word count. Each of the documents in the corpus is represented by columns of equal length. Those are wordcount vectors, an output stripped of context.

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

Before they're fed to the neural network, each vector of word counts is normalized such that all elements of the vector add up to one. Thus, the frequency of each word is effectively converted to represent the probabilities of those words' occurrence in the document. Probabilities that surpass certain levels will activate nodes in the network and influence the document's classification.

2. Tf-IDf: It is another way to judge the topic of an article by the words it contains. With TF-IDF, words are given weight – TF-IDF measures relevance, not frequency. That is, wordcounts are replaced with TF-IDF scores across the whole dataset. First, TF-IDF measures the number of times that words appear in a given document (that's "term frequency"). But because words such as "and" or "the" appear frequently in all documents, those must be systematically discounted. That's the inverse-document frequency part. The more documents a word appears in, the less valuable that word is as a signal to differentiate any given document. That's intended to leave only the frequent AND distinctive words as markers. Each word's TF-IDF relevance is a normalized data format that also adds up to one.

$$w_{i,j} = tf_{i,j} \times log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Those marker words are then fed to the neural net as features in order to determine the topic covered by the document that contains them.

Following this we train our LDA model using gensim.models.LdaMulticore to obtain the topics. LDA refers to the latent Dirichlet allocation which is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics.

Experiments and Results

Forecasting

We have chosen 9 countries for analysis. These are the 8 most affected countries and India. The countries are namely US, China, Spain, Italy, France, Germany, Iran and Singapore. The countries are considered most affected based on the number of confirmed cases and number of deaths due to Covid-19.

1. The first algorithm that was chosen was Elastic net. It is a hybrid of Lasso and Ridge, where both the absolute value penalization and squared penalization are included, being regulated with another coefficient l1 ratio: $\frac{1}{2m}\sum_{i=1}^{m}(y-Xw)^2+\alpha(ratio)\sum_{j=1}^{p}w_j+\frac{1}{2}\alpha(1-ratio)\sum_{j=1}^{p}w_j^2$

In our experiment, we tried the elastic net on 3 different data manipulations.

-The first model is built on a data that has the target variable as the cumulative number of confirmed cases/fatalities up to a particular date

-The second model predicts the number of new cases on any given date.

-The third model calculates the weekly average number of new cases.

The reason we tried these 3 different variations is, calculating cumulative numbers in general incurs lower errors, since the cumulative numbers are generally bigger than the daily new numbers. If there are any differences from this assumption in the error numbers, that can lead us to interesting insights. We also tried weekly average because, the error rate is mathematically smaller when predicting the moving average.

2. The second algorithm we implemented was Random forest regression. In this case, based on the results we got from the 2 experiments with elastic net, moving averages seemed to be the best approach to go with. We experimented

moving average for a range of intervals (3-7 days) and predictions of 3-day average number of cases showed the least error and hence we have included the results of the same in this paper.

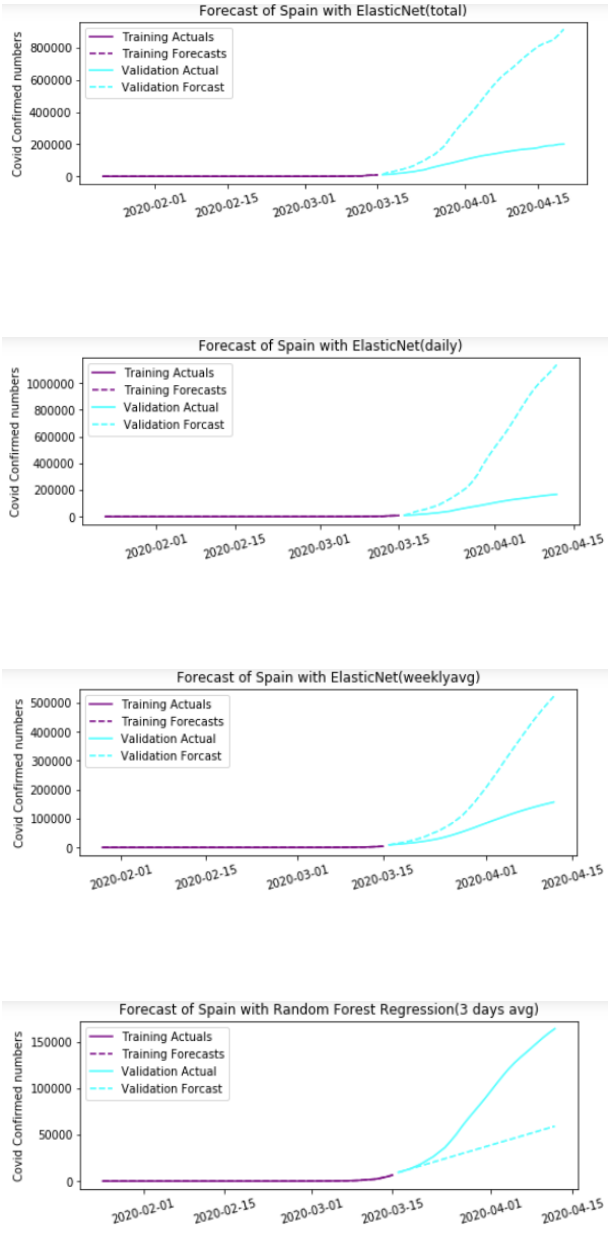


Fig1. The images above show the predictions for Spain for three different elastic net implementations and one random forest regression. The purple lines indicate the training data and the blue lines indicate the validation data. The continuous lines indicate the actual values and the dotted lines indicate the predictions of the models.

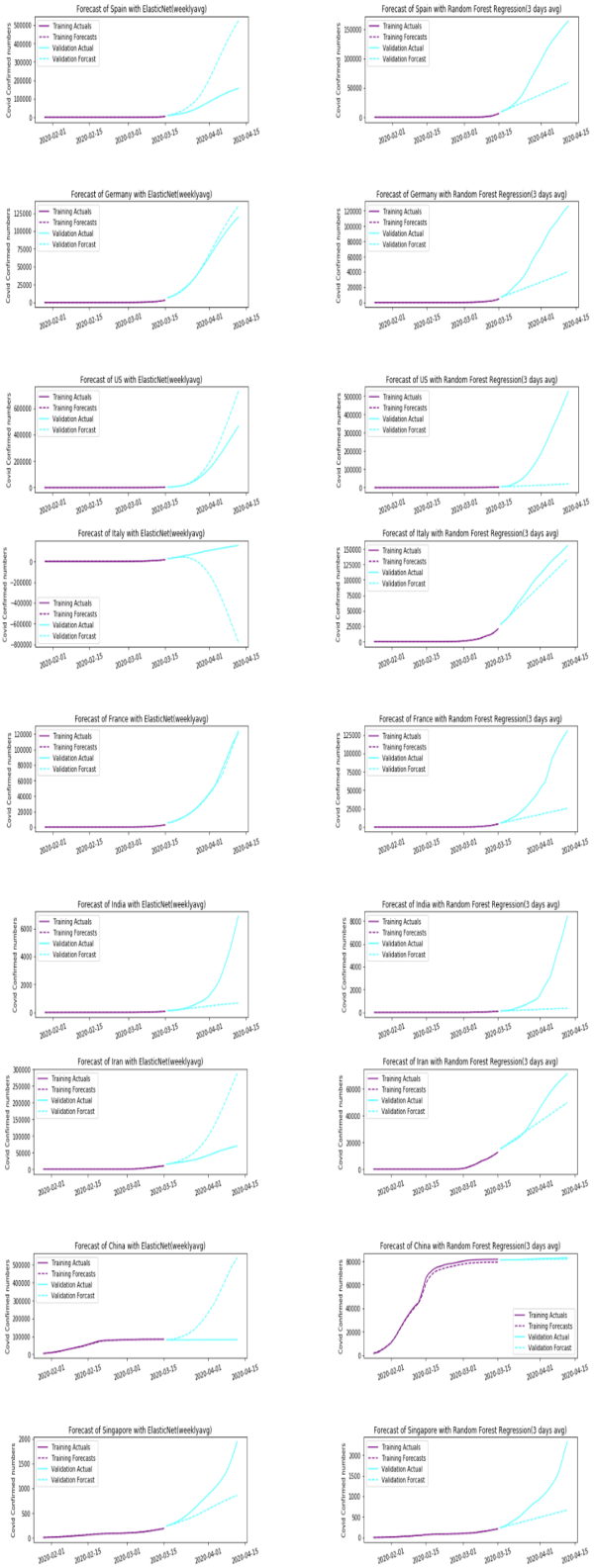


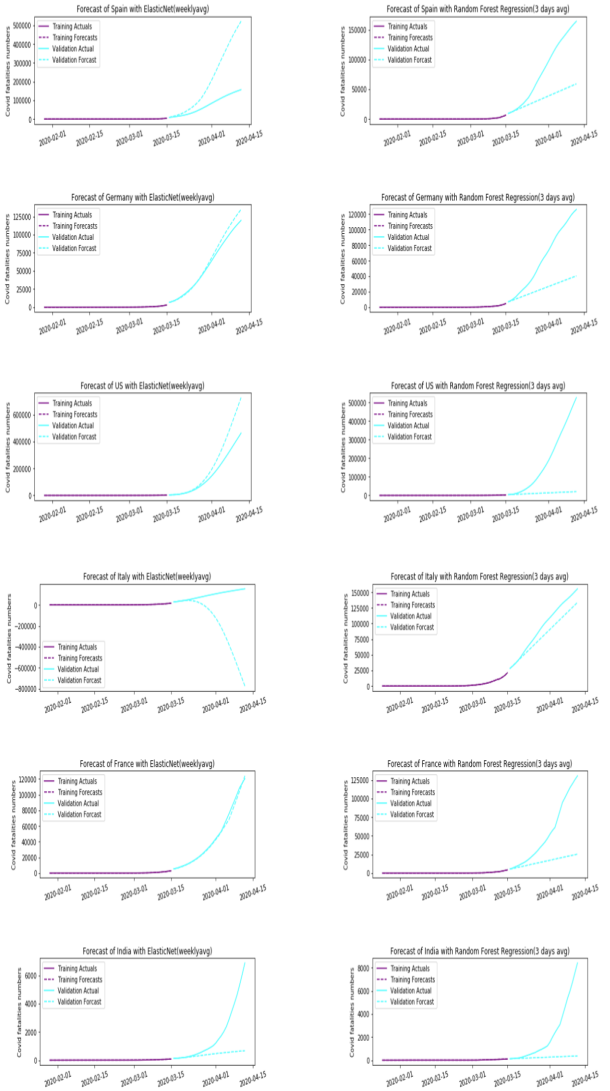
Fig2. Since, out of the 3 different Elastic-net implementations, the weekly average predictions show lower errors in most cases, we have illustrated the outputs of Elastic-net weekly average and corresponding Random-forest auto-

regression predictions for the number of confirmed cases for each of the countries.

RMSLE\ Countries	ElasticNet (total)	ElasticNet (daily)	ElasticNet (weekly avg)	RF Regression (3 days avg)
Spain	1.209	1.578	0.942	0.817
Germany	0.113	0.054	0.088	0.921
US	0.297	0.559	0.330	2.569
Italy	0.655	0.522	1.039	0.134
France	0.474	1.559	0.042	1.114
India	2.338	0.436	1.639	2.285
Iran	1.408	0.771	1.024	0.266
China	0.916	0.852	1.285	0.013
Singapore	1.276	0.537	0.710	1.007

Fig3. This is the table of RMSLE errors for the estimated number of confirmed cases for the 3 variations of Elastic-net predictions and a Random Forest Regression.

The results displayed above are for the predictions of number of confirmed cases. We executed the exact same procedures on the data for number of fatalities.



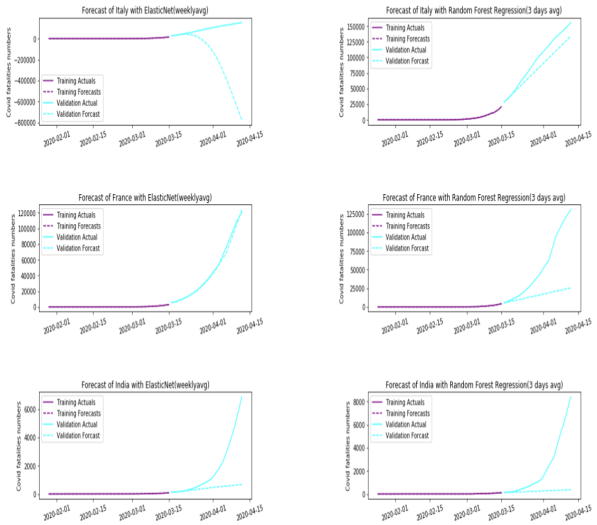


Fig4. Outputs of Elastic-net (weekly average) and corresponding Random-forest auto-regression results for number of fatalities for each of the countries.

RMSLE\ Countries	ElasticNet (total)	ElasticNet (daily)	ElasticNet (weekly avg)	RF Regression (3 days avg)
Spain	1.307	1.682	1.014	1.541
Germany	1.381	1.296	1.191	2.782
US	0.123	0.321	0.119	3.385
Italy	2.030	0.638	0.286	0.483
France	2.252	0.586	0.643	2.549
India	1.415	2.097	1.165	1.887
Iran	0.239	0.336	0.088	0.210
China	0.057	0.218	0.174	0.059
Singapore	1.725	1.725	1.568	1.673

Fig5. These are the RMSLE errors for the estimated number of fatalities for the 3 variations of Elastic-net predictions and a Random Forest Regression.

From the results on the validation set, there is no clear indication of a best model across all the countries. For example, Elastic-net seems to be doing much better than Random-forest regression for Germany, US and France. However, Random-forest does better than Elastic-net for China, Italy and Iran. Hence from the results above we can not conclude on particular algorithm that is best suited for all the countries. The best approach will be to use different algorithms for each of the countries based on the performance on validation set.

Parameters we used to run the Elastic-net algorithm are alpha:np.logspace(-4,2,10), l1 ratio = np.array([0.6,0.7,0.8,0.9,1.]), where: "Alpha" is constant that multiplies the penalty terms. Defaults to 1.0. See the notes for the exact mathematical meaning of this parameter. alpha = 0 is equivalent to an ordinary least square, solved by the LinearRegression object. For numerical reasons, using alpha = 0 with the Lasso object is not advised. Given this, you should use the LinearRegression object, l1 ratio float is ElasticNet mixing parameter, with 0 lesser than or equal to l1 ratio lesser than or equal to 1. For

l1 ratio = 0 the penalty is an L2 penalty. For l1 ratio = 1 it is an L1 penalty. For 0 lesser than l1 ratio lesser than 1, the penalty is a combination of L1 and L2.

Parameters we used to run the Random Forest Regression are n estimators:[10,20,30], max features : [”auto”, ”sqrt”, ”log2”], min samples split : [2,4,8], where: n estimators is the number of trees in the forest, max features is the number of features to consider when looking for the best split, min samples split is the the minimum number of samples required to split an internal node.

We also collected some additional data from external sources such as World Bank to investigate variables that might be influencing the spread of the pandemic. We plotted a simple correlation matrix of the confirmed cases and fatalities with other variables such as population density, land area, number of migrants, mortality rate, lock down date,yearly number of flight passengers etc.

Fig6. Correlation matrix where the shades tending to red scales indicate negative correlation, shades tending to blue scales indicate positive correlation and the intensity of the contour indicates the extent of correlation.

Topic Modeling

Topic modeling is done using LDA on corpus built using bag of words and another after applying Tf-idf. We have used gensim model LdaMulticore on python for topic modelling.

Of the topics generated using bag of words corpus we did identified two topics out of 10 being about Chinese. These topics included considerable number of negative words like ”stupid” which did not occur in other topics that were not about Chinese, hence showing that there are a lot of conversations with negative sentiments around the Chinese population or the Chinese government.

The most talked about topic that emerged using Tf-idf processed corpus consists of words like Starbucks since Wuhan’s starbucks re-opening was viral on social media. It also has high focus on employment and unemployment followed by words like bills, payments etc. which ties strongly to the struggles the middle class people of the society are going through.

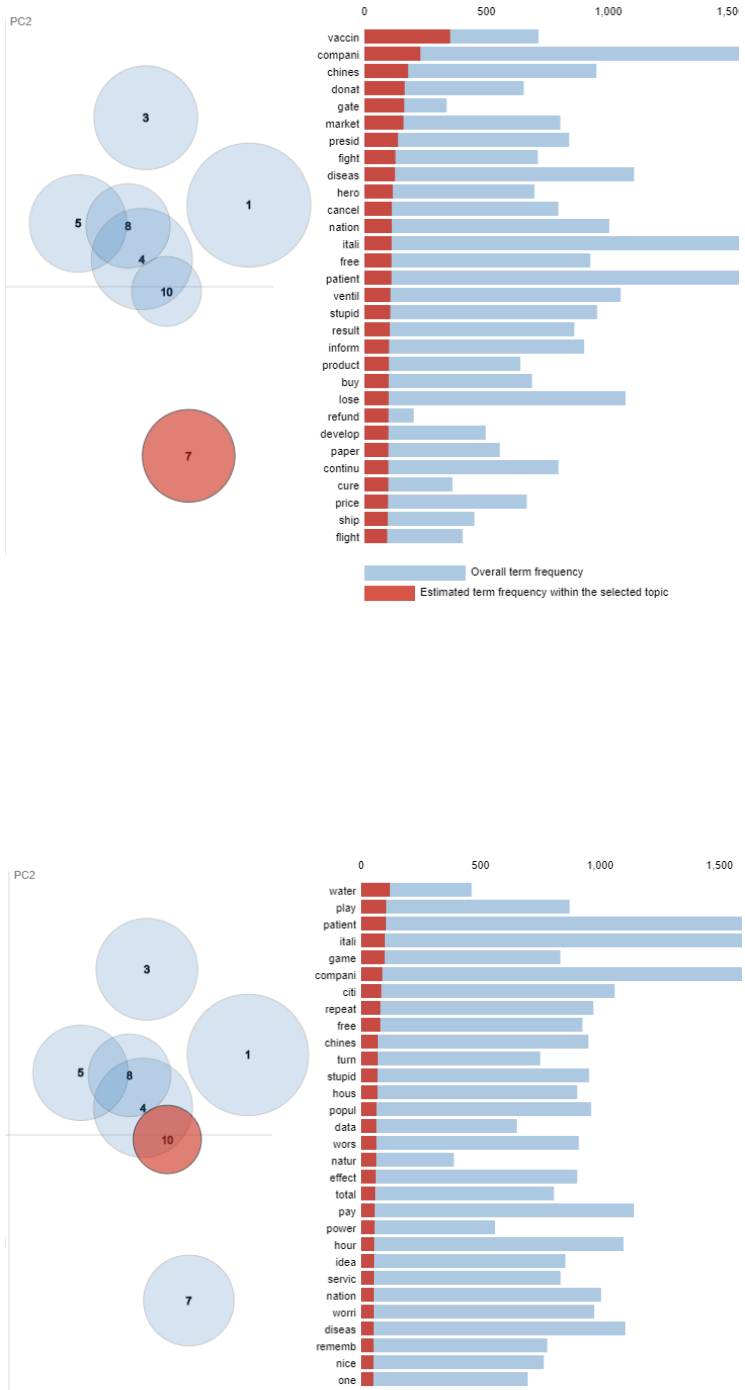


Fig7. These are the results of topic modeling, highlighting two topics that revolve around words like ”Chinese” and ”Stupid” occurring together.

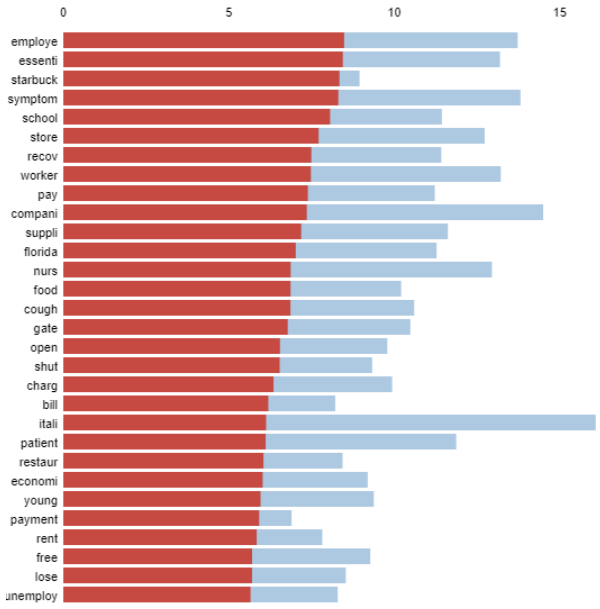


Fig8. This is the LDA result using the corpus processed with Tf-idf. This image highlights the most talked about topics.

We also plotted a word cloud consisting of most used monograms.

Fig9. This word cloud is built on cleaned text from Reddit to depict the most frequently used words in the conversations.

Conclusion

Forecasting an epidemic at this age is a noble but extremely difficult task. From the perspective of machine learning the quest to find the most accurate model with very limited data has attracted a lot of attention. There are a lot of data augmentation methods and bootstrapping methods to improve the performance of the models (Fong et al.(2020)Fong, Li, Dey, Gonzalez Crespo, and Herrera-Viedma). In this paper we have implemented two auto regressive methods and fine tuned the parameters to give the best results with the algorithms. We saw the two different algorithms we tried, have proved to be suitable for different countries. It is hard to choose one algorithm that is doing better than the other for all the countries. Despite the fact that our experiment demonstrates the possibility of picking a suitable forecasting model for each of these countries, we need to be aware that forecasting is a dynamic task whose predicted result is highly sensitive to the parameters used, the choice of model and the training data. Any addition or modification of the training data with the arrival of new data, is likely to influence the course of the predicted result.

From a machine learning point of view, it is important to test more algorithms for panel selection. It would be even better if we probe the underlying designs of suitable algorithms and understand the why each of them incurs very low or very high errors. In reality, pondering on how exactly an epidemic is going to propagate is a complex and wide multifactor decision. Basing the propagation solely on historical data is not enough since it is only one of the many factors, though important. A more thorough methodology would be to simulate the trends by modeling the factors that tend to affect the transmission and enhance the model iteratively as more data becomes available.

As far as social media is considered, it can provide a great insight on what is the biggest worry or need of the common population that has access to internet. It can serve to be beneficial for the governments to understand their public better in crisis situations like these. Social media also has the power to influence people's minds in a socially and racially discriminating manner which will most likely be detrimental to the society during a pandemic. It might lead to innocent individual belonging to certain races being discriminated and being denied of services in critical times. Hence we have a responsibility to curb down strong sentiments against races on social media during times like these when humanity needs to fight against the problem as a whole.

References

John S Brownstein, Clark C Freifeld, and Lawrence C Madoff. Digital disease detection—harnessing the web for public health surveillance. *The New England journal of medicine*, 360(21):2153, 2009.

Simon Fong, Gloria Li, Nilanjan Dey, Ruben Gonzalez Crespo, and Enrique Herrera-Viedma. Finding an accurate early forecasting model from small dataset: A case of 2019-ncov novel coronavirus outbreak. *International Journal of Interactive Multimedia and Artificial Intelligence*, 6:132–140, 03 2020. doi: 10.9781/ijimai.2020.02.002.

Lin Jia, Kewen Li, Yu Jiang, Xin Guo, et al. Prediction and analysis of coronavirus disease 2019. *arXiv preprint arXiv:2003.05447*, 2020.

Gabriel J Milinovich, Ricardo J Soares Magalhães, and Wenbiao Hu. Role of big data in the early detection of ebola and other emerging infectious diseases. *The Lancet Global Health*, 3(1):e20–e21, 2015.