Data Analysis of COVID-19 Patients in South Korea

Songrui He
SID: 862188777

Yi Wang
SID: 862186591

Yiyuan Liu
SID: 862188392

Yuxuan Liu
SID: 862188304

Guoren Li
SID: 862188830

Abstract—**To date, COVID-19 has become one of the most serious human pandemics and has spread for more than 6 months. Identifying the susceptible people and taking preventive measures accordingly is the key point to defeat the disease. Based on the reported materials of KCDC, this paper adapted and evaluated several machine learning algorithms to predict the growth trend of new confirmed cases and identify patients with high death risks. In addition, graph mining techniques are used to analyze group infection and super infector cases. These methods and analysis provided instructive information to help tackle the global problem of Coronavirus.**

Keywords—COVID-19, Machine Learning, Data Mining, Linear Regression, SVM, Logistic Regression, Naïve Bayes, K-means, Graph Mining

I. Introduction

Coronavirus disease 2019 (COVID-19) is the infectious disease caused by the most recently discovered coronavirus. The disease has spread globally from the ongoing 2019–2020 coronavirus pandemic. As of 8 June 2020, more than 7 million cases have been reported in 210 countries and territories, resulting in more than 400,000 deaths. Governments of many countries and CDC recommended people to stay at home to prevent COVID-19.

We want to make a contribution. We used the structured dataset based on the report materials of KCDC and local governments [1] because this dataset is detailed enough for us to analyze and mine. It is known that identifying the death risk of all patients is critical for a country/region to control the disease, allocate the medical resources and take some preventive actions. According to existing researches, there are some relationship between the patient's condition(e.g. Age, sex, underlying disease, country/region) and their death risks. Besides, the spread of the virus is influenced by the region, population, migration etc. based on a hidden model.

In this project, by using data mining algorithms and methods, we predicted the growth trend of new confirmed cases in a particular region, identified groups of people with high death risk and classified patients' state based on the given information of the patients. Besides, we use graph mining techniques to detect the hidden relations of patients. In summary, we make the following contributions:

1) Use linear regression to predict new confirmed cases in the future 2) Use SVM, Naïve Bayes and Logistic Regression to predict the state of patients based on selected features. 3) Use k-means to analyze hidden relations of groups of patients. 4) Use graph mining techniques to analyze group infection and super infectors.

The remainder of this paper is organized as follows. Section II provides some related work. Section III presents the preprocessing work. Section IV provides the details of the algorithms/methods that we used. Section V shows the experimental result and evaluation. Section VI concludes the paper.

II. Related Work

David et al. [2] proposed k-means++ algorithm based on traditional k-means method. By augmenting k-means with a simple, randomized seeding technique, k-means++ improves both the speed and the accuracy. Flake[3] told the effect of SMO algorithm when using SVM. And he discussed both pros and cons of SVM's SMO algorithm. From this paper, we learned the basic conception of SMO algorithm and how to make its advantage bigger. Kevin [4] Murphy introduced and summarized classical Naïve Bayes classifiers. In his work, we are able to know the Multinomial Naïve Bayes algorithm is suitable for our current work (e.g. classify a patient, predict whether he is in high risk of COVID-19).

Siddiqui et al. [5] found the relationship between temperature and different COVID-2019 cases. By using K-means clustering algorithm, they found that temperature may have some relationship with COVID 2019 cases situation, but the relationship is not significant, which means temperature may not be the main factor. Du et al. [6] employs the Hybrid AI Model to make predictions. Compared to traditional models, Hybrid AI model is more effective and can significantly reduce the errors of the prediction results. Gaurav et al. [7] aim at finding the rate of spread of the disease in India, developing SEIR (Susceptible, Exposed, Infectious, Recovered) model to evaluate the spread of disease. They used Regression models to predict COVID-19 outbreak, and turn factors to susceptible, exposes, infectious, recovered degrees, then build a mathematical SEIR model. Finally SEIR model has good performance and the prediction result has high accuracy. Dianbo Liu et al. [8] is trying to provide stable and accurate COVID-19 forecasts ahead of days. Their paper presents a novel methodology which combines mechanistic model and machine learning methodologies to forecast COVID-19 2 days ahead of current time at the province level in China. In particular, it's based on and also improves the latest and most accurate infectious disease prediction model - ARGONet. Sajadi et al. [9] uses the dataset provided by EAR5 to analyze and compare the climate of the places where COVID-19 spreads more seriously and the climate of less serious places.

Yang et al. [10] uses an improved SEIR model to integrate virus transmission trends in three provinces (Guangdong, Hubei, and Zhejiang) in China. In addition, the optimized LSTM model is used to predict the number of infected cases. Dehning, Jonas, et al. [11] proposed a method that combining an established epidemiological model, Susceptible-Infected-Recovered (SIR), with Bayesian inference to analyze the time dependence of the effective growth rate of new infections. The results quantify the effect of interventions, and the corresponding change points can be incorporated into forecasts of future scenarios and case numbers.

III. Data Preprocessing

OpenRefine[12] and scikit-learn[13] is used to do some data preprocessing work. The work mainly consists of the following three parts.

A. Data Reduction

We deleted some useless and redundant columns in our dataset, which is shown in Table 1.

Deleted Column	Reason
"latitude" & "longitude" in "Case.csv"	Province and city will be used as location information.
"time" in "Time.csv"	Data update time is not useful for prediction

Table 1

B. Dealing with missing data

In PatientInfo.csv, we did the following processing works:
1) There are 77 records missing "sex", "age" and "infected_by" values. We treated them as "lacking too much information" and deleted them.
2) For the records missing "sex"(there are only 9 records), since the rest of the records shows the ratio of "female" to "male" is 1.27:1, we manually filled in the missing values with 5 female and 4 male in random.
3) There are 7 records missing both "age" and "birth_year". We filled in with a new category "unknown" in "age" for these records. For the rest of the records missing either "age" or "birth_year", we can calculate one of them based on the other value.
4) For the records missing "city" values, since we know the province, we filled in "city" with existing cities in that province based on the proportion of existing cities.
5) For the rest missing values, we filled in with a new category "unknown".

The rest of the dataset are processed based on similar strategies as listed below.

C. Data Transformation

To make it easier to analyze and visualize, we encoded the categorical features using LabelEncoder from scikit-learn. After that, we change the range of features values to make standardize the data after encoding, using MaxAbsScaler from scikit-learn.

IV. Proposed Method

A. Correlation Analysis

To find the relations quickly and efficiently between different features in our dataset, the most effective way is to perform correlation analysis on these random variables.

The first analysis is to confirm the relationship between the number of patients and whether they are grouped or not. Because this is a highly dangerous new virus that can be passed on from person to person, when people form groups in public, the chance of getting sick should be great. To verify this idea, the relevant bar plot is constructed based on python.

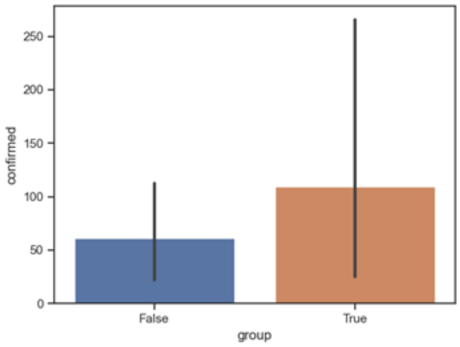

Fig. 1

From Fig. 1 we can see that most of the infected patients have experienced grouping with others before.

Based on the results of a paper mentioned in my related work, we tried to find out whether there is a linear relationship between the number of diagnoses in COVID-19 Korea and latitude and longitude. To do that, first we need to process the data. To avoid interference from unrelated factors, case_id and group variables are deleted. With the help of Open Refine, the longitude and latitude variables in the dataset are set to float. Then we drew the joint plot regression graph of longitude and number of patients and latitude and number of patients respectively in Fig. 2 and Fig. 3.

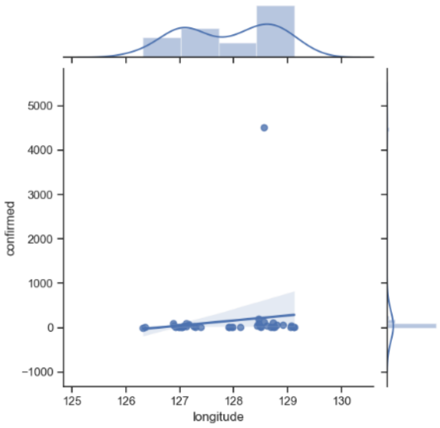

Fig. 2

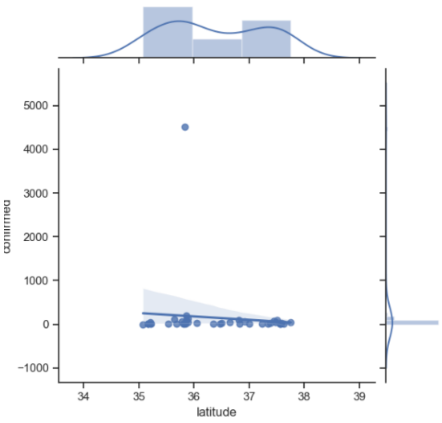

Fig. 3

We can see that because there are almost 5000 people infected in one place, the influence of latitude and longitude on it seems to be minimal. To further explore the relationship between latitude and longitude and confirmed cases, we need to calculate the correlation coefficient between them, a heat map is shown in Fig. 4.

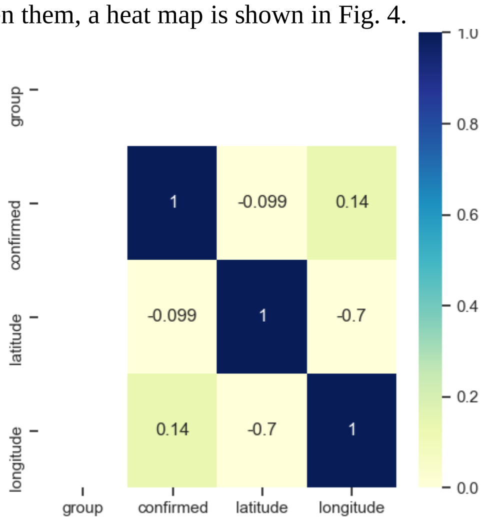

Fig. 4

Unfortunately, based on the correlation coefficient, both longitude and latitude are rarely related to the number of infections. we personally think that this is because South Korea is not a country with a large latitude and longitude. If we use dataset from China or the United States to test, the results may be more in line with our expectations. After that, we tried to find out the correlation between the four variables of patient status, age, gender and health status. So, we process the data, label the patient’s status by three numbers (0: released, 1: isolated, 2: deceased), removed all data except these four columns, and draw heat map based on this:

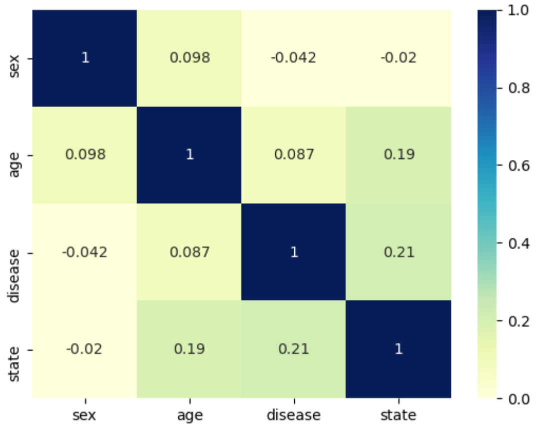

Fig. 5

Among these four variables, the correlation coefficient of disease and state is the largest, which is expected.

B. Linear Regression

Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).

In our project, we used Ridge Regression Model for analyzing multiple regression data that suffer from multicollinearity. Fig. 6 shows loss function and weights of ridge regression. Ridge regression adds a regularize part dominated by lambda to mitigate the least squared part. To make the outcome of loss function to be minimum, the gradient of L should be zero. Then we can get the weights of ridge regression as showed below.

$$L = \sum_{i=1}^{m} (y_i - w^\top x_i)^2 + \lambda \underbrace{w^\top w}_{\|w\|_2^2}$$

$$w_{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top Y$$

C. Logistic Regression

We established a logistic regression model to predict whether a patient infected with COVID-19 will die.

First, the essence of regression is to fit a bunch of data points with a straight line. Among them, logistic regression is to establish a new regression formula according to the classification boundary line in the existing data, and use this formula to classify other data[14].

The prediction function we generate is the Sigmoid function. In order to consider the state that the decision boundary is indistinguishable with a one-dimensional line, we can expand the boundary form to obtain our prediction function.

$$h_\theta(x) = sigmoid(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

For the binary classification problem, the output value of the prediction function here is greater than 0.5, and the category is 1, otherwise it is 0.

To find the θ vector here, we need the Cost function and the J(θ) function (The specific derivation process is omitted here):

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

$$cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if y=1} \\ -log(1 - h_\theta(x)) & \text{if y=0} \end{cases}$$

Then we can use the gradient descent method to find the minimum value of J(θ), so that we can classify.

In our project, due to logistic regression is more suitable for binary classification, so we need to modify the origin dataset to 2 classes (e.g. ‘isolated’ patients and ‘deceased’ patients can be somewhat combined to same class because they both in high risk of COVID-19).

Logistic regression needs extra parameters like step size, iterate times. We simply assign them as 0.05 and 1000.

D. Naïve Bayes Classification

Naïve Bayes Classification is a classical and feasible classification model.

In our project, due to the discrete distribution data, we used Multinomial Naïve Bayes Classification algorithm, its posterior probability function is described as follows:

$$argmax_{c_k} logP(Y = c_k) + \sum_{j=1}^{d} x^{(j)} logP(w_j | Y = c_k)$$

In this algorithm, our mainly task is to calculate out the prior probabilities of classes, and the mean and variance values of features. Then use these variables to compute the posterior probability of new data point, then pick the label with highest probability as its prediction result.

We selected 5 features from dataset, include: sex, age, country, province, infection case.

E. SVM

Support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis[14].

In our project, we used SMO(Sequential Minimal Optimization) algorithm to implement our linear SVM method. This SVM method can build a model which can classify different states by providing patient information.

And to make our SVM classifier can classify multi-class, we used one-versus-one(OVO) method. This method pick any two classes and use one classifier to classify them, so it builds k(k-1)/2 SVM classifier when there are k classes. In our case, we have three different states which refers to three different classes. So, we build three different SVM classifier: A, B and C. In test phase, every record will be voted by all the three classifiers.

(A,B)-classifier: if A win, A=A+1. Otherwise, B=B+1;
(A,C)-classifier: if A win, A=A+1. Otherwise, C=C+1;
(B,C)-classifier: if B win, B=B+1. Otherwise, C=C+1;
The decision is the Max(A,B,C).

We selected 6 features from dataset, include: sex, age, country, province, infection case and infection order.

F. K-means

K-means algorithm is one of the unsupervised cluster methods. It clusters data in n groups by minimizing the sum of squares(SSE). For example, there are N samples $(X_0, \dots X_n)$ need to be divided into K disjoint clusters C. Each cluster is described by the mean of the samples in this cluster: μ_j, which is commonly called "centroids". The SSE function is described as follows:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

K-means algorithm requires the number of clusters K to be specified. To get the optimal K, the algorithm is executed several times and SSE is calculated under different K.

We use 7 features to perform the K-means algorithm: sex, age, country, province, city, disease, infection_case. And the optimal K is set to 3.

G. Graph Mining

Based on the 'patient_id' and 'infected_by" features in the patient info dataset, a graph can be constructed to connect people with "infection relation". We use NetworkX[15] to generate the graph with nodes and edges.

Group Infection Detection: Group infection normally consists of several people having strong infection relations with each other. To obtain the potential group infection cases(like a community), we used "Community Detection"[16] , which is based on louvain method described in "Fast unfolding of communities in large networks".

Super Infector Detection: A "super infector" is defined as someone spreading the virus to a large amount of people. Based on the community detection, we used an API of NetworkX "betweenness_centrality" to get the nodes' centrality and compute the mean centrality within a community(similar to PageRank). Based on the rank, if a node's centrality is higher than the threshold we set, it is treated as super infector.

V. Experimental Evaluation

A. New Comfirmed Cases Predction

Yellow line is real confirmed cases while blue line is the prediction of confirmed cases. The train set factors include temperature, humidity and covid-19 search trend. We trained data from first 60% days, and predicted the confirmed cases in later 40% days. Fig. 6 shows the result.

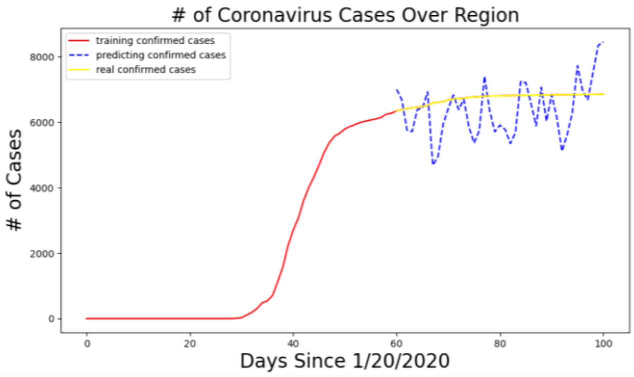

Fig. 6

B. Identify High Death Risk Patient

1) Naïve Bayes:

When we tried to implement Naïve Bayes with original preprocessed data, we found that the accuracy is extremely low, which is about 12%, even worse than just guess.

Interestingly, later we found we neither implemented unsuitable Naïve Bayes Algorithm, nor preprocessed the dataset correctly. At the beginning, we chose Gaussian Naïve Bayes algorithm and resulted in poor performance, with deep researching we found that the dataset is actually discrete distribution, which means there's no explicit comparison of each feature values, so we need to implement the Multinomial Naïve Bayes Algorithm, which is much more suitable for such dataset. In the patient information dataset, we have a feature 'infection order', which is used for tracking COVID-19 infection. But due to some unknown reasons, about 98% data of this feature is blank, which means there are too much missing data in it, which will cause the data distribution not as our expect to implement neither Naïve Bayes nor Logistic Regression. Then we pick out this useless feature and have a reasonable result.

We randomly split the dataset as 4:1, ran 50 times, and got the average accuracy 73.16%. Compared with sklearn-package's Multinomial Naïve Bayes Classifier, which is about 74.97% accuracy, our classifier's performance is reasonable.

2) SVM:

Though there are many null input data in some features, the SVM method still got lowest 71.6012%, highest 95.18072% and average about 84.73% accuracy. That means SVM is an appropriate method to analyze this data and it is insensitive to null inputs or it shows that some features are abnormal cannot infect final result too much.

Though linear SVM gains a not bad performance in our project, we suggest whether non-linear SVM can do better.

And we can also try to use gradient descent algorithm to build SVM in the future.

3) Logistic Regression:

We tried logistic regression with different step sizes and different iteration numbers.

Step	Accuracy
0.2	65.7%
0.1	67.1%
0.05	71.3%

Table 2

In the step size test, we set the iteration times as 1000 and ran 50 times of test.

Iteration	Accuracy
10	61.9%
100	70.2%
1000	71.3%

Table 3

In the iteration time test, we set the step size as 0.05 and ran 50 times of test.

Ideally logistic regression should also concern about automatically find out best step size and most efficient iteration times. The balance between cost and accuracy in this model is interesting, gradient descent makes the model will never overfit dataset.

The three classification algorithms are compared after running 10 times separately. The result is shown in Fig.7.

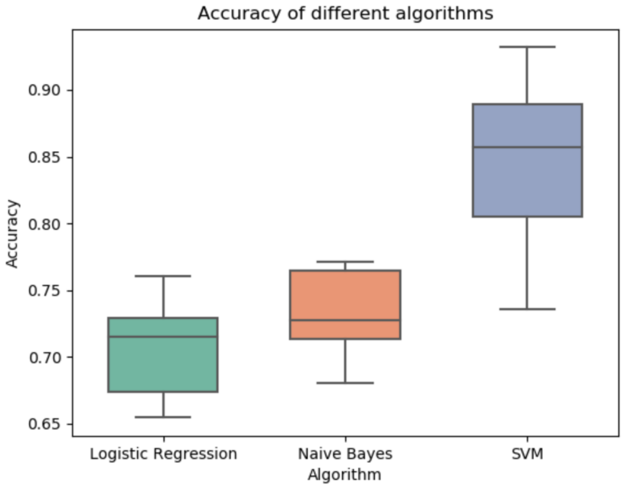

Fig. 7

4) K-means:

All people are divided into three clusters based on K-means. Death rate is defined as (the number of death in this cluster) / (total number of people in this cluster). The clusters details are shown in the following table.

	Amount	Death rate
cluster1	1879	0.3%
cluster2	1142	0.8%
cluster3	283	14%

Table. 4

The results show that people died of COVID-19(deceased cases) do have hidden relations, which means particular groups of people(cluster3) have high death risks.

Further analysis shows that this group of people comes from Korea or United States. 84% of the people in this group have underlying diseases before caught on COVID-19. And the infection case of this group mainly lies on 'overseas inflow' (25.1%) and 'Suyeong-gu Kindergarten' (39.4%). The age of this group mainly lies on 60~80(70%).

C. Graph Mining

Group infection detection graph is shown as follows. Different colors means different communities(infection groups). There are 11 major group infection cases as shown in Fig. 8.

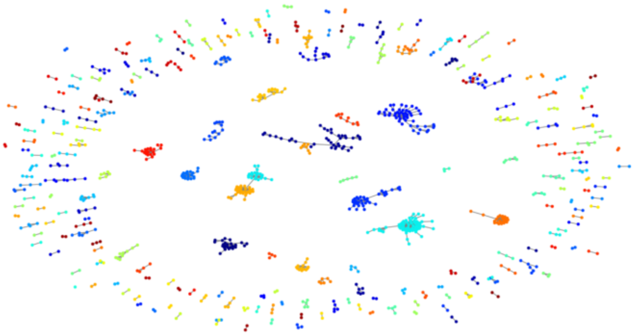

Fig. 8

Details of certain major communities are clearer after zooming in Fig.9.

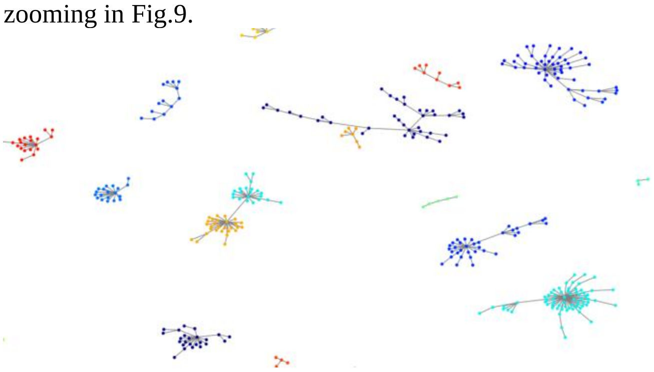

Fig. 9

After adding the centrality rank, super infectors are found as shown in Fig 10.

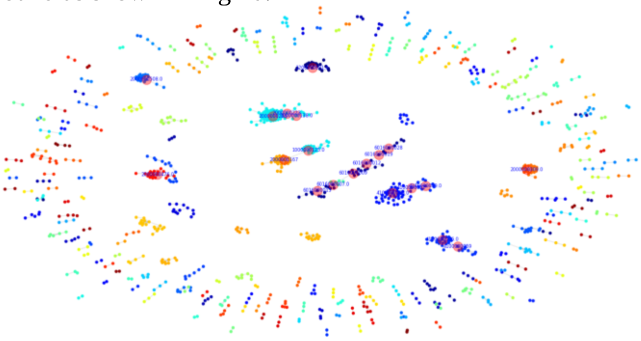

Fig. 10

Fig 11 and Fig 12 shows some major super infectors after zooming in.

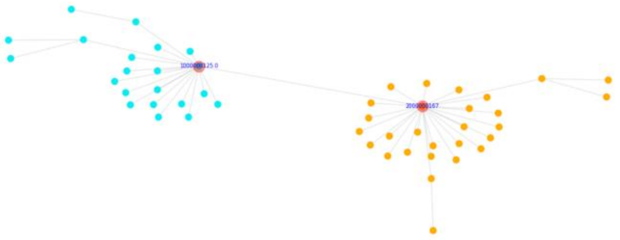

Fig. 11

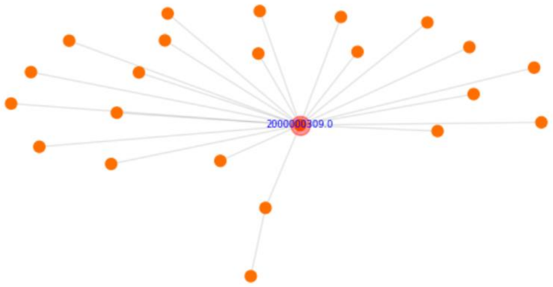

Fig. 12

According to the graph, a list of super infectors' patient ID can be obtained: [6016000015, 6016000009, 6016000012, 6016000028, 4100000022, 4100000049, 4100000059, 1000000125, 1000000138, 6016000007, 6016000019, 1200000031, 2000000167, 2000000205, 2000000223, 2000000309, 2000000476, 2000000508, 4100000006, 4100000008]

VI. Discussion & Conclusions

The result of predicting new confirmed cases shows that some acceptable bias and to get this result, we only used three features: "coronavirus" search trend, humidity and temperature. That may reveal the main factors of infected COVID-19 are humidity, temperature and virus search trend which looks reasonable because humidity and temperature are influencing people's immunity and survival of virus, search trend can show the degree of caring from people.

The result of predicting states of patient implies that the final states of patient may influence by gender, age, province, country and infection case. These also sound reasonable. However, there may be more inherent reasons which we do not mine from the data, such as policy, city and so on. As shown in the result of clustering analysis, people from Korea and United States with underlying diseases before and age from 60 to 80 have higher death rate than other people. This result can help CDC to assign and provide health resources effectively.

By using graph mining techniques, we found super infector and infection group clearly. In the future, we can analyze more from this method and we believe it can help many people to prevent COVID-19.

For the total four different task showed above, we both get relatively good accuracy. We think one of the reasons is our methods are appropriate to this dataset and we also preprocess data in an appropriate way. However, we may do it better by improving the methods to process and analyze data and we still have amount of things waiting for mining.

References

[1] https://www.kaggle.com/kimjihoo/coronavirusdataset

[2] Arthur, David, and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Stanford, 2006.

[3] Flake, Gary William, and Steve Lawrence. "Efficient SVM regression training with SMO." Machine Learning 46.1-3 (2002): 271-290.

[4] Murphy, Kevin P. "Naive bayes classifiers." University of British Columbia 18 (2006): 60.

[5] Siddiqui, Mohammad Khubeb, et al. "Correlation Between Temperature and COVID-19 (Suspected, Confirmed and Death) Cases based on Machine Learning Analysis." J. Pure Appl. Microbiol 14 (2020).

[6] Du, Shaoyi, et al. "Predicting COVID-19 Using Hybrid AI Model." (2020).

[7] Gaurav Pandey, Poonam Chaudhary, Rajan Gupta, Saibal Pal. "SEIR and Regression Model based COVID-19 outbreak predictions in India." MedRxiv (2020)

[8] Liu, Dianbo, et al. "A machine learning methodology for real-time forecasting of the 2019-2020 COVID-19 outbreak using Internet searches, news alerts, and estimates from mechanistic models." arXiv preprint arXiv:2004.04019 (2020).

[9] Sajadi, Mohammad M., et al. "Temperature and latitude analysis to predict potential spread and seasonality for COVID-19." Available at SSRN 3550308 (2020).

[10] Yang, Zifeng, et al. "Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions." Journal of Thoracic Disease 12.3 (2020): 165.

[11] Dehning, Jonas, et al. "Inferring COVID-19 spreading rates and potential change points for case number forecasts." arXiv preprint arXiv:2004.01105 (2020).

[12] https://openrefine.org/

[13] https://scikit-learn.org/stable/

[14] https://en.wikipedia.org/wiki/Support_vector_machine

[15] https://networkx.github.io/documentation/networkx-1.9/index.html

[16] https://python-louvain.readthedocs.io/en/latest/