Prediction of Covid-19 Cases in Ohio State

Puhua Ye

***Abstract*— Using a training data set which including Covid-19 variables, awareness level variables and county-level variables for different counties in Ohio state builds a prediction model. Predict the number of Covid-19 cases for different counties. Use methods including data pre-processing, machine learning, etc.**

I. INTRODUCTION

In this project, I am going to predict number of Covd-19 cases on a given day in a county of Ohio based on level of awareness for various topics, including core, domestic issues / domestic politics, economy, education, gender, health, and etc, and many county-level variables, such as total population of the county, percentage of the people who have at lest finished high school, and etc. Given the original training data set, I did some transformation on variables to make the data meaningful and easily classified during training model. Also, because of the big amount of attributes, I did dimensionality reduction according to the meaning of different variables. After that, according to different algorithms that I implemented, I used different data per-processing techniques to prepare the data for training. In order to reach the highest accuracy, I implemented different machine learning algorithms, including support vector regressor, decision tree regressor and light gradient boosting machine. Among three models, the highest accuracy reached by LightGBM model which is 87.921%.

II. DESCRIPTIVE DATA ANALYSIS

A. Training Data.CVS

This data set includes 3141 observations. For each observation, it contains 144 attributes, which mainly consists of 3 parts. One is county-level data, such as deaths of a given day, total-population of a county, and etc. One is awareness-level data, measured by different similarity measures(Jaccard, Cosine, Intersection), which extracted from tweets and has been collected by using hashtags, that including "name of the pandemic / Covid", posted by users in Ohio during Covid-19 pandemic. The last part is the awareness-level of data that 0-1 normalization has been used. Also, the "cases" variable in the training data is the target variable that I am going to predict.

B. Test Data.CVS

This data set also include 7331 observations. For each observation, it contains the same 144 attributes with the training data set, except the target variable, "cases".

C. Visualization

First, I plot a bar chart that shows the average normalized Jaccard similarity-based awareness value for different topics and order the bar chart from the biggest to the smallest.

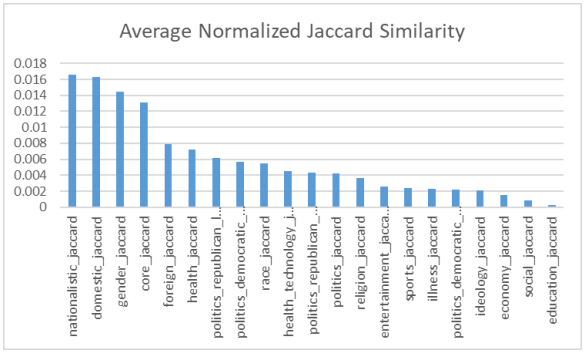
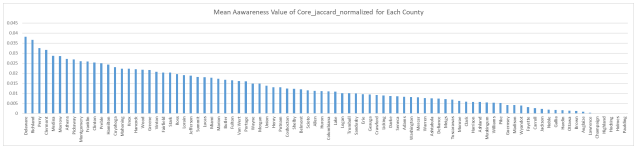


Fig. 1. Average Normalized Jaccard Similarity

From the graph, I found that the most topics that related to Covid-19 measured by Jaccard similarity is the nationalistic topic. The similarity of domestic topic is nearly equal to the nationalistic topic. And the top four topic is nationalistic, domestic, gender and core. The Jaccard Similarity of all those four topic are above 0.01, and have a large lead compared with other topics.

Second, I created a bar chart that shows the aggregated mean awareness value of core-jaccard-normalized for each county and order the bar chart from the biggest to the smallest.



From the graph, I found that Delaware county publish the most tweets that are related to Covid-19. And I found that, for the counties that with the top-15 average awareness value of core-jaccard-normalized. also has most Covid-19 cases, which shows that those two variables has some correlation.

Third, I created two county-level maps and used colors to show the number of average Covid-19 cases per capita and the number of average Covid-19 deaths per capita.

From the graph, the top-5 counties with high number of per capita cases are Franklin, Coyahoga, Pickaway, Lucas and Hamilton and the top-5 counties with high number of per-capita deaths are Cuyahoga, Lucas, Summit, Franklin,

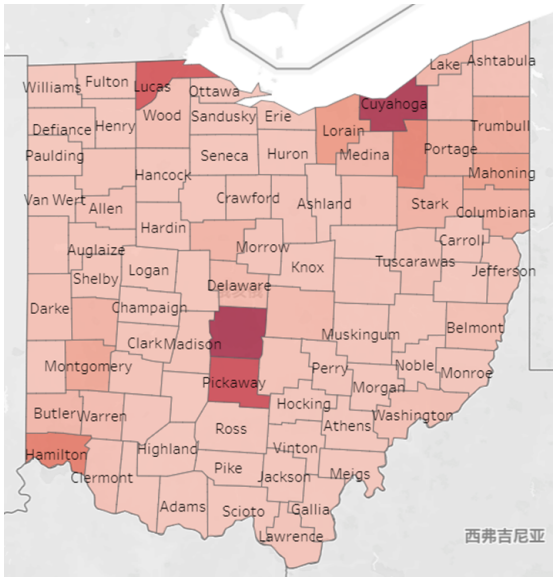


Fig. 3. Number of Average Covid-19 Cases per Capita

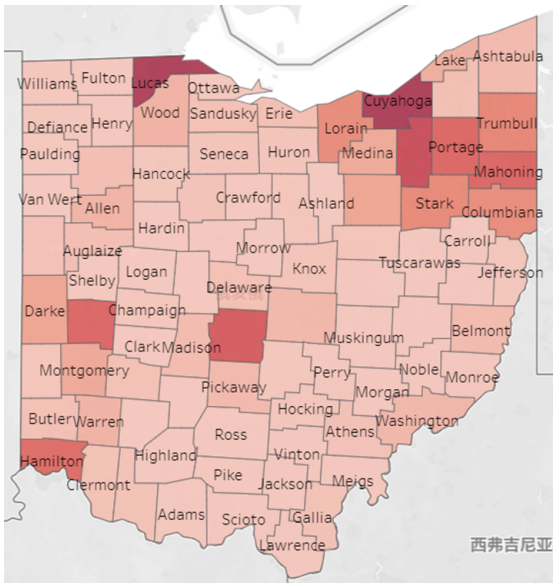


Fig. 4. Number of Average Covid-19 Deaths per Capita

and Mahoning. The comparison between number of per-capita cases and number of per-capita deaths shows that there is a strong correlation between cases and deaths. Therefore, "death" is a important variable to help predict the number of cases.

Forth, I created a line chart with overlapping lines in which each line represents the evolution of awareness levels for each topic.

In the graph, X-axis represents "Day" variable, and Y-axis represent the level of awareness. The overall data values are all under 0.05, and on some special days, there is a substantial increase in the AWARENESS level for certain topics. This may be directly related to the hot events or special activities of the day.

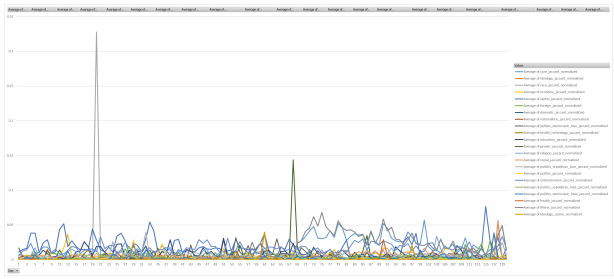


Fig. 5. Evolution of Awareness Level for Each Topic

III. Methods

A. Data Pre-processing

During data pre-processing, I first transform the date-index-converted to variable "day". Through this, we can predict the number of Covid-19 cases from the perspective of time-series, which is a very important variable in our prediction model. Moreover, in the original data set, there are two awareness level variable for the same topic. One is non-normalized, and another is a normalized variable. Because I used LightGBM algorithm which is based on the tree structure, I did not use the normalized variables. Thus, I deleted those awareness level attributes that are normalized to reduce the dimensionality of data set. Also, in the LightGBM algorithm, it can directly process categorical variable. So, I change the type of "county" variable to the categorical type. Because the county is a categorical variable, the county-level variable is not necessary to used to distinguish different counties. So, I deleted those county-level variables to further reduce the dimensionality. After all data pre-processing steps, I have a data set with 27 attributes, which is much less than the 144 attributes in the original data set. Here is a lift for final training data variables:

```
Index(['cases', 'county', 'deaths', 'Day', 'core_jaccard', 'core_cosine',
       'core_intersection', 'social_jaccard', 'politics_jaccard',
       'politics_democratic_love_jaccard', 'nationalistic_jaccard',
       'politics_republican_hate_jaccard', 'entertainment_jaccard',
       'sports_jaccard', 'race_jaccard', 'economy_jaccard', 'foreign_jaccard',
       'religion_jaccard', 'health_jaccard',
       'politics_republican_love_jaccard', 'health_technology_jaccard',
       'politics_democratic_hate_jaccard', 'domestic_jaccard',
       'illness_jaccard', 'ideology_jaccard', 'education_jaccard',
       'gender_jaccard'],
      dtype='object')
```

Fig. 6. Predictor Variables

B. Model

At the beginning, because this is a prediction problem, I select Support Vector Regressor to solve the problem. However, the final accuracy of SVR is not great, no matter what kernel I used. Then, I selected to use Decision Tree Regressor because I draw multiple graphs between target variable and other variables and did not find clear linear or polynomial relationship, so I think I can use a classification ideas to separate different situations and to predict the number of cases. After implementing Decision Tree Regressor, the accuracy has improved to 80%. After

that, I tried to find more advanced algorithm which use decision tree as base. So, I learned about LightGBM, which is a high-performance gradient boosting algorithm that uses decision trees to improve predictive accuracy. Compared to the Decision Tree Regressor, LightGBM has higher accuracy and shorter training time, but it has more hyper-parameter need to be tuned.

C. Hyper-Parameter Tuning

In order to get a higher-performance, it is necessary to tune the hyper-parameter of LightGBM. The key parameter of LightGBM includes learning-rate, max-depth, min-leaves, and number of iterations. I tried different combination of those parameters and use MSE and R2 score as metrics to avoid over-fitting to find the best combinations.

D. Prediction

Because, from the stage of descriptive analysis of data, I found there is almost no cases before day 70. Therefore, I change the final predictions that resulted from LightGBM model of day before day 70 to 0, in order to get a better accuracy.

IV. Result

The final accuracy of my model is 87.774%. For the LightGBM model, I used plot-importance function from LightGBM to see which variables are the most important.

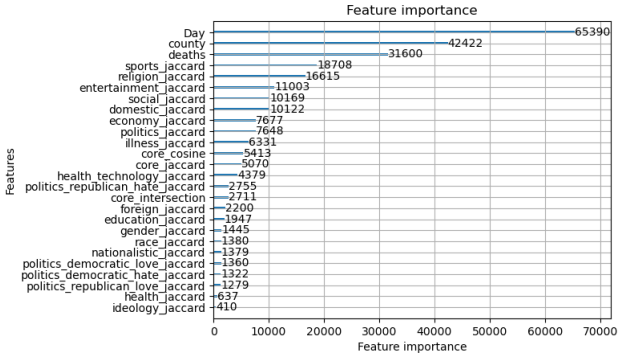


Fig. 7. Evolution of Awareness Level for Each Topic

From the graph, we can find that the top-3 important variables are Day, County and Deaths.

During the process of project, I tried lots of different models and pre-processing methods, such as standard scaler, min-max scaler. Many algorithms did not result a good accuracy. It also made me reflect that the most important thing in data analysis may not be the use of more advanced models, but the cleaning and processing of data according to the actual situation in order to get a higher accuracy rate.

V. Conclusion

In this project, I did different steps of data cleaning and data-pre-processing to prepare a good data set for training model. Then I tried different regression algorithms to find the best model which result the highest accuracy. Finally, I build a LightGBM model with fitting hyper-parameter by training on training.csv. Then, I use the model to predict the number of Covid-19 cases for different counties of Ohio in a given day. With the final accuracy of 87.921%. I believe that my model works well on predicting the number of cases of Covid-19 for Ohio State.