DS100: Data Analysis and Prediction on COVID19 Dataset

Star Li, Zheng Zhang, Stefan Li
(University of California, Berkeley)
May 14, 2020

1 Problem Introduction

With over a million confirmed cases of COVID-19 in the United States, we have witnessed waves of stock market meltdowns, growing unemployment populations, and shortage of critical resources like ventilators. Consequently, an urgent question at this moment is: what will happen next? Answering this question would require the ability to accurately predict the number of confirmed cases in certain regions. Understanding the trend of the virus's spread is critical to the allocation of different resources and the enactment of government policies. Furthermore, we also filtered out the data from other countries, such as China, since our main focus is predicting cases in the U.S. In this project, we want to apply data science workflow to the prediction of confirmed cases in U.S. counties (specificially at 4/18). We start from the time series data, trying to fit an exponential model as the basis of our prediction. Then we focus on incorporating more data using a blend of automatic feature selection and manual feature engineering. Inspired by our EDA visualization, we also tried to inegrate KNN algorithm improves our model by bringing data of nearby places. Our final model is a combination of linear regression and regression tree, which successfully reduces the overall MSE below 50 and performs extremely well on regions with low confirm cases.

2 Data Cleaning

2.1 Date Selection and Overall Procedure

Based on our objective of predicting confirmed cases in the U.S., we selected the following three provide dataframs:

- **4.18states**: contains specific demographic and medical information by STATE.
- **time series covid19 confirmed US**: this table contains the most import time series information regarding the confirmed cases by county and state.
- **abridged couties**: Contains a even more detailed population demographic, medical and health info by county and state.

Overall Procedure:

Since **4.18states** is the only by-state dataframe, we first cleaned up all the NaN values and unwanted values of it. Then, since **time series covid19 confirmed US** and **abridged counties** are both by state and country, we merged these two table with "UID" of time serise table and "countyFIPS" of abridged counties as our key for merging the data. Lastly, we combined all three data together in order to provide a raw Xtrain matrix called **all in 1** with all the NaNs cleaned up and edge cases removed.

2.2 Cleaning up "4.18states"

Filter out special cases:
Notice that there are rows such as "Diamond Princess", "Grand Princess", and "Recovered" in the dataframe. Since we only care about regional predictions, these special cases are irrelevant to our goal and keeping these data might be misleading for fitting our model. We also filter out the rows of whose "Last Update" columns are NaN. If the data is not up-to-date, it might negatively impact our prediction.
Filling in NaN values:
As we inspected our data with Pandas codes, we found out that columns that contains NaNs are: **'Recovered', 'Active', 'Mortality Rate', 'Hospitalization Rate', 'People Hospitalized'**

- Based on the provided README file, we notice that **Active cases = total confirmed - total recovered -**

total deaths, so we can safely fill out all the NaN's in Deaths, Recorvered, and Active with 0, since they are all mutually exclusive.

- Notice that US Hospitalization Rate $= \frac{\text{Total number hospitalized}}{\text{number confirmed cases}}$, so if the number of People Hospitalized is NaN or 0, we can logically fill in 0 for all the NaN's in these two columns
- We end up cleaning up all the NaN values in 4.8states after performing the above procedures.

2.3 Join "time series covid19 confirmed US" and "abridged counties"

Filter Out Special Cases:
Since we only want to focus on the prediction in the states within the U.S., we filtered out some territories of the U.S., such as "Puerto Rico " and "Guam", since they are geographically far away from the main 50 states and lacking of detailed information in both tables.
Join two tables:
we merged these two table with "UID" of time serise table and "countyFIPS + 84000000" (they are off by 84000000 for every state) of abridged counties as our key for merging the data.
Filling in NaN values:
There are lots of NaN values in some of the columns, in order to give a consistent and relatively accurate X matrix for fitting our models, we decided to drop out the columns with too many NaNs, since if there are too many NaNs in a category, it would be really hard to choose alternative for them since demographic information are very unique to each state, and filling up all the NaN data with "mean" "median" or "most frequent" will potentially bring misleading information to our models. To define **"too many"**, we decided that if more than **20 percent** of the data are NaN, we will drop it.

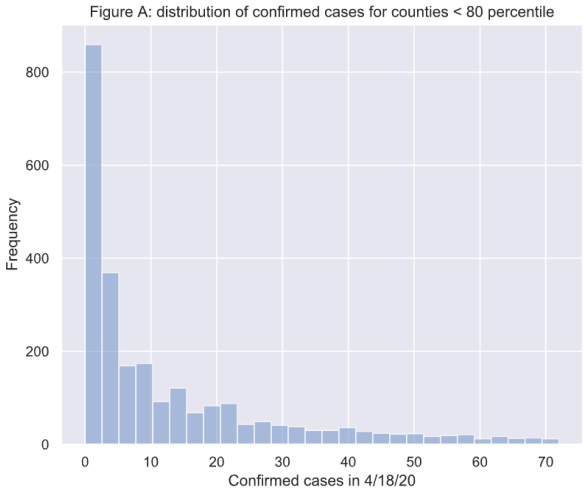
After filter out the ones with a lot of NaN values, we have the following columns with NaN values, and I explained how we deal with each kind of NaN columns with specific reasoning:

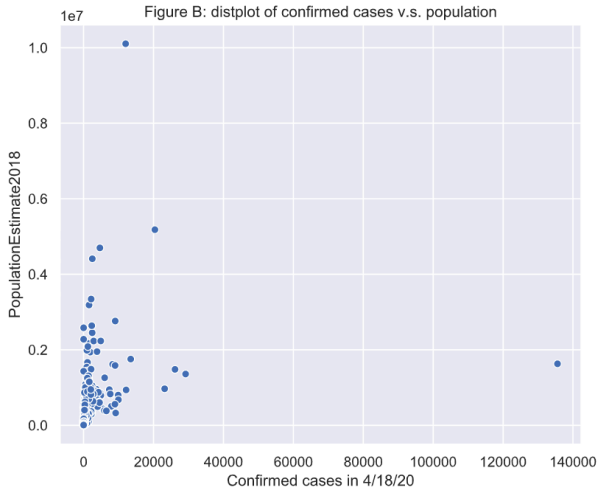
- Columns to Drop Directly:
  1. **State**: The "Province State" from the time series table has no NaN values, so "Province State" can already represent the names of all the State.
  2. **lat, lon**: The Lat and lon from the time series table has no NaN values, so it can present all the locations
  3. **entertainment/gym**: We think gym and entertainment might have long-term effect for human body, but it is not very related to the virus that is happening right now.
- Columns to take the "Mean" value to fill NaNs:
  1. **EligibleforMedicare2018**: Since it has such a small NaN percentage and Medicare system is relatively well-developed in the U.S., we decided to fill the mean for the NaN of this column
  2. **All the rate, ratio, and percentile**: Since rate and ratio has already scaled, we can safely apply rate and ratio to states and counties with different populations.
  3. **All the MortalityAge**: Since all of them have such low NaN precentage, we decided to fill in Mean for them, as Mean will average out the effect of states with large population and small population.
- Columns to take the "0" value to fill NaNs:
  1. **>50 and >500 gatherings**: since gatherings might have a huge impact on confirmed cases, we rather to do it more safely by filling in 0s for these features with NaNs.

At the end, we merged the two tables above and created the All in 1 table for model and feature selection.

3 Exploratory Data Analysis and Feature Selection Rationals

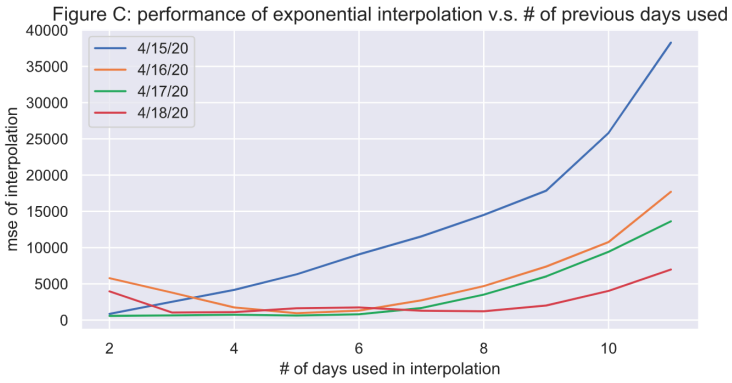
3.1 Distribution of Confirmed Cases





The first step in our EDA is to grasp the overall distribution of our target – confirmed cases up to 4/18/20. In figure A, we plot out the distribution of confirmed cases for counties within lower 80 percentiles. It shows how left-skewed the distribution is as 80% of counties have less than 80 cases and more than $\frac{1}{3}$ of counties have less than 10. Figure B gives us a more complete view of the distribution of populations and confirmed cases, as we can see the clusters of points near the origin and some "outlier-like" points far away, which helps us greatly in the downstream model selection. We also didn't find a strong correlation between population size and the confirmed case.

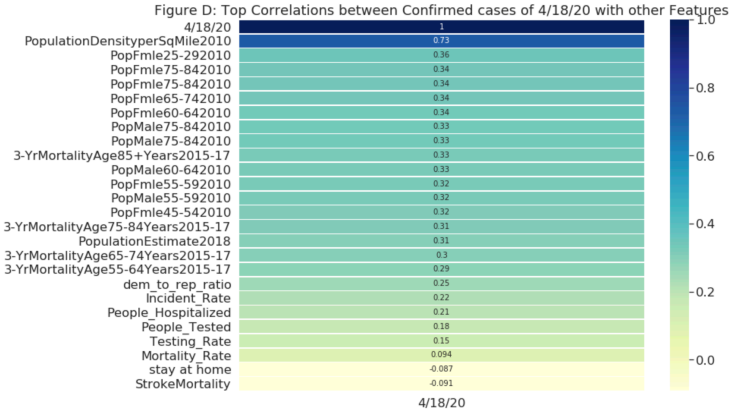
3.2 Time series: which days matter?



Our first assumption follows from the common exponential model for epidemic growth [1]. We take advantage of this and try to answer "which days in the time series can accurately predict the future" using exponential interpolation. More specifically, to generalize this assumption and reduce our search space, we fit exponential curves for "n consecutive days earlier" ($n \in [2, 11]$) (e.g. if we predict cases in 4/16 using $n = 2$, we fit a curve using 4/14 and 4/15's cases) and measure their interpolation ability with MSE. The above graph shows the result when we apply this procedure to the prediction from 4/15 to 4/18 (but only the 4/18 one is our real interest). All curves experience increase in MSE as we incorporate more previous days in the exponential fitting; cases in 4/18 is predicted the best if we fit $n = 3$ (using 4/15, 4/16, 4/17's cases).

[1] 3B1B has made a good video on the intuition behind this model https://www.youtube.com/watch?v=Kas0tIxDvrg

3.3 Visualizations for Selected Features



Based on the results of using Lasso-based feature selector and ExtraTreesClassifier, we created a heatmap visualization to show the correlations between the selected features and confirmed cases of 04/18. We did not show other time series features cause those are clearly correlate with 04/18. Notice that population demographics, especially population density and Female population segment, have relatively strong correlations with confirmed cases of 04/18, while features that seem to correlate with the virus, such as People Hospitalized, Testing Rate, and Incident Rate, have relatively low correlations. This also make sense as based on the algorithm of how these two selector works: Lasso-based feature selector will try to minimize absolute loss function when selecting features and ExtraTreesClassifier tries to find the most distinguishable features for each level of confirmed quantity, which make Population density an excellent feature that is distinguishable among different state and counties. [2]

4 Model and Experiments

4.1 Feature Selection

This process is a blend of manual feature selection and automatic subset selection.

Manual Feature Selection: we first take advantage of the results from our EDA. For time series data, we follow the results in Figure C, which shows that confirmed cases in the past 3 days give the best prediction on 4/18's number (It's worth noting that we didn't explicitly feature engineer the interpolated number for 4/18 using the fitted exponential curve since we believe our regressor will do this implicitly as long as the features are offered). We also include the features which have relatively high correlation with the confirmed cases using our heatmap in Figure D. In addition, we have done some research on the internet and several articles about diabetes[3] and blue versus red states [4]drew our attention.

Automatic Subset Selection: To discover potential useful features, we apply the Lasso regression to perform automatic subset selection. As we have learnt, the additional l_1 norm of Lasso regression keeps the weights of unimportant features to be close to 0. This gives us the features **dem-to-rep-ratio** and **PopulationDensityperSqMile2010**, where the former is consistent with the aforementioned article. We originally tried to use tree-based feature selection methods such as ExtraTreeRegressor and AdaBoostRegressor; however, they suffer from high variance and cannot give stable evaluations of the features. Problems with tree-based regressors will be discussed in the model selection part.

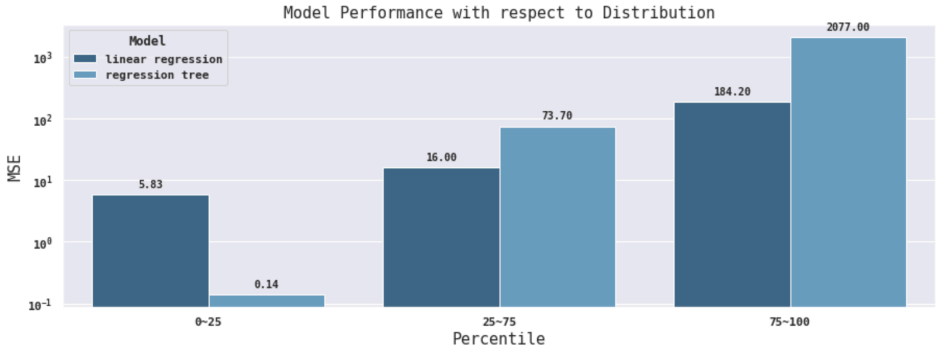
The above parts give us the following subset of features: **{4/15, 4/16, 4/17, dem-to-rep-ratio, public schools, FracMale2017, DiabetesPercentage, People-Tested, HeartDiseaseMortality, PopulationDensityperSqMile2010}** for use in the following model selection part.

[2]SKlearn documentation about how each algorithm works https://scikit-learn.org/stable/modules/feature$_s$election.html
[3]https://www.touchendocrinology.com/insight/covid-19-infection-in-people-with-diabetes/
[4]https://www.latimes.com/politics/newsletter/2020-05-08/covid-hits-red-states-essential-politics

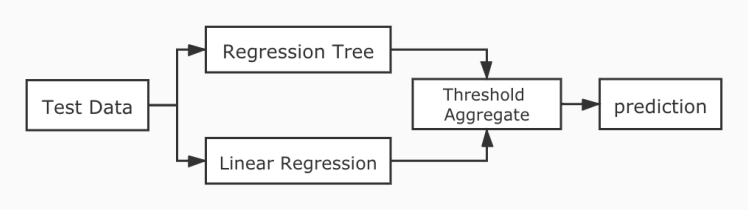
4.2 Model Selection

Our problem is a regression problem and we use MSE to measure the performance of our model. We start out using simple linear regression and decision tree regression (or regression tree) as prototypes. For regression tree we use cross-validation to tune the hyperparameter maximum depth. A very interesting and useful finding is that linear regression works better for counties with large confirmed cases while the regression tree does the opposite. This is reflected from the summary statistics below, where the numbers are the MSEs for predicting confirmed cases in different percentiles.

	0-25 percentile	25-75 percentile	75-100 percentile
linear regression	5.83	16	184.2
regression tree	0.14	73.7	2077



To explain this observation, we should look back to Figure A and B in EDA, which show that the confirmed cases are right-skewed with most of the data centering around 0. For the regression tree, since it tries to approximate some curve (underlying function of the model) with decision rules[5], it tends to give finer approximation where the data are denser (i.e. small numbers in our case); but it can only give a course regression when data are sparse such as the few "outliers" in Figure B. On the other hand, the linear method are more sensitive to a few large numbers of confirmed cases at the expense of sacrificing some precision in predicting small numbers.

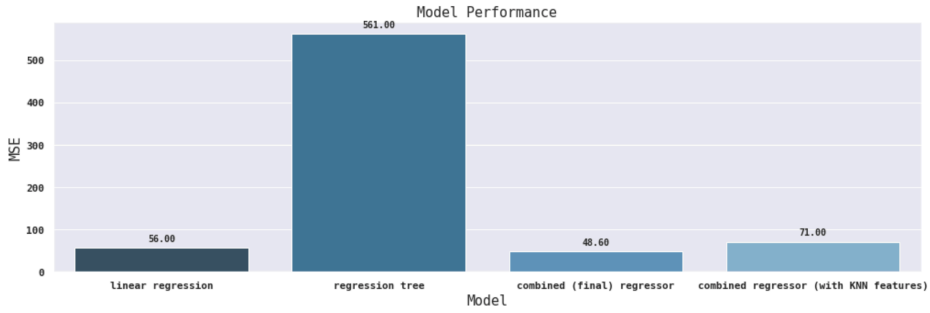


Our final model thus combines both regressors and achieves lower MSE by taking advantage of our observation. Intuitively, we train both a linear regressor and a regression tree in the training phase. We also try to find a percentile cutoff using the training data (confirmed cases), which serve as a threshold for making prediction. Here the percentile is a hyperparameter that we have tuned using cross-validation on the training set. In the prediction part, for each sample, we give two predictions with our two regressors, then taking an average of the two. If the average is below our threshold, we choose the regressor tree prediction and we use the linear regression result otherwise.

We've also tried K-Nearest-Neighbor regressor to find the geographically adjacent counties of a certain county, and aggregate their features like confirmed cases, population, and hospital numbers for prediction. This didn't work quite well since the variance of confirmed cases among the counties are very high, and the KNN regressor performs poorly for counties that are very "different" from its neighbors. The results of all the approaches we have tried can be summarized by the table below.

	linear regressor	regression tree	combined (final) regressor	combined regressor (with KNN)
MSE	56	561	48.6	71

[5]Here the sklearn gives a very nice picture of this approximation https://scikit-learn.org/stable/modules/tree.htmltree



5 Analysis and Conclusion

5.1 Final Result

Our final model imitates the boosting method in statistical learning, combining two sub-optimal regressors into a better one by giving them each some "votes". In our case, we vote for the regression tree when the number predicted is small, and for linear regression when it's big – a natural choice following our observation and experimentation. In the end, our final model lowers the overall prediction MSE below 50 using only 9 features, only 3 of which are time series data. For places with less than 100 confirmed cases, which is more than 80 percent of counties, our prediction MSE is lower than 2.

5.2 7 Questions Responses

1. There are some very interesting features that we found such as **"public schools", "FracMale2017", and "DiabetesPercentage"**. As we previously thought features that relates to number of gatherings and Incident rate would have huge impact on predictions, these three surprising features actually opened our vision and motivate us to find more features that aren't seem to be intuitive correlated with the confirmed cases.
2. Based both on the Heatmap we created as well as the feature selected from other selector, it is very interesting that "stay at home" has such a low impact on the number of confirmed cases prediction. In fact, as we add this variable into our model, our MSE increased. As we thought "Stay at Home" might lead to a huge impact on decreasing the number of confirmed case, it actually has negative impact on our model. This is partially because the variance of this feature is not that high across all the counties.
3. The most challenging task are probability to clean up the data set as well as choosing the right model for it. In terms of data cleaning, there are many NaN values in all the data set and it is very important to find the right method to deal with nulls as it might bring unpredictable effect to our prediction. In terms of model selection, we really get stuck when our Regression Tree is very good at prediction places with low confirmed cases but bad at places with high cases. It took us a while to find out this results and a lot of observations for us to figure out how to combine the benefits of both Regression Tree and Linear Regression to better fit with our data set.
4. **Limitations:** Although finding a cutoff to decide which model to use provide us a relatively good results, depending only on the percentile of confirmed cases is probably not be the most optimal way to do since there are many other features that we can potentially use for specifying our model. Also, our prediction relies heavily on the confirmed cases from the previous 3 days, so our prediction is very sensitive to the authenticity of these prior data; Finally, our model might not work for more up-to-date data since it doesn't take into account factors that would undermine our assumption of an exponential growth model (e.g. social distancing, shelter-in-place policies). **Assumptions:** We made a lot of assumptions when cleaning up our data, such as filling up rates with mean, dropping the columns with more than 20 percent of null values etc. However, features from the provided data set are extremely unique for each state and county and these NaN data could highly correlate with a region's medical record or other information that are relevant to Covid-19. The way we clean up our data might disable us from finding more useful features to fit our model.
5. There are some sensitive information appear when we are working with data such as "dem-to-rep-ratio" and how female's population proportion have much higher correlations with confirmed cases as oppose to male's.

The first dilemmas is politically sensitive as our results might indicate that one party is better at dealing with Covid-19 than the other one. As for female proportion, our data might also indicate that regions with more female are more vulnerable to the spread of Covid-19.

6. A more up-to-date version of data set that could fill up all the NaN values for population demographic will provide use with more confidence of our prediction as most of population data are from 2018 or even earlier. Also, another data set with more medical records, such as number of pneumonia cases, will provide features that have higher correlation with our predicted values. Furthermore, a data set can indicate inter-state communications will even better as we will be able to analyze how neighborhood states could impact the number of confirmed cases.
7. We are concerned about how our results might lead to certain kind of sexual or political discrimination in a region. As our results might indicates that political ratio and female population have relatively high correlations with confirmed cases of Covid-19. Our results might become a misleading source in regional election for one party to impost disadvantage to the other. Thus, to deal with these issues, we will address these concerns with further studies, and **explicitly state that since our data set is not comprehensive, the impact of female population and political demographic might not be statistically significant**, and one should not infer any conclusion without further statistical analysis. Also, the fact that "Stay at home" is not highly correlated with confirmed cases might also be misleading, as Stay at home order is critical for controlling the virus all around the world, and the fact that our model does not put much weights on this feature dose not imply the importance of stay at home order.

5.3 Future Work

Even though our current research has made some impressive results, there are still many spaces for improvement in the future.

1. *Explore new methods:* neural networks, especially RNN (LSTM), would be a good fit to our time series data. We would like to apply deep learning techniques if we have more time.
2. *Refine current methods:* the KNN regressor didn't work well in our research partly because we didn't carefully select the distance metric and aggregation method. A weighted linear combination of features from neighbors might give us better results than simple averaging.
3. *Integrate extra features:* if we have more time, we would also like to include extra features on the Internet that don't exist in the dataset given to us. As we have mentioned above, a more up-to-date population estimate for each county might help us give more accurate prediction.