CIS 5200 Fall 2022, Project Report

Predicting COVID-19 Cases From County Health Data

John-Wesley Appleton
johnwes@wharton.upenn.edu

Abstract

An accurate count of new daily cases of COVID-19 is important for resource allocation and policy making at the county, state, and federal levels. We sought to provide an accurate prediction of new daily cases of COVID-19 at the county-level for those counties which do not have the resources nor the facilities to test and report the actual number of new daily cases. We explored the use of various machine learning methods to predict new daily cases of COVID-19 at the county-level in 2020.

1 Motivation

COVID-19 is a coronavirus disease discovered in December 2019 [2]. It is very contagious and spreads through droplets that you project out of your mouth or nose when you breathe, cough, sneeze, or speak. On January 31, 2020, the World Health Organization (WHO) issued a global health emergency, and on March 11, 2020, WHO declared COVID-19 a global pandemic [1]. On June 10, 2020, COVID-19 cases in the United States reached 2 million, and on August 17, 2020, it became the third-leading cause of death in the US [1].

Obtaining an accurate count of for COVID-19 cases has always posed a challenge; towards the beginning of the pandemic the challenge stemmed from a lack of testing resources [4] and recently, from a lack of self-reporting with self-test kits [10].

Machine learning can assist in the prediction of daily cases by leveraging both previous COVID-19 data and county health factors. By considering all the information available, we expect to form relatively accurate predictions which could prove beneficial to counties across the United State.

2 Related Work

Given that COVID-19 is a coronavirus disease, there has been substantial research examining the short-term and long-term effects of COVID-19 on the numerous physical and biological systems. Additionally, there has also been extensive research examining the lasting political and economical impacts of this global pandemic. For the purposes of this report, we are most interested in related work in attempting to predict the count of daily cases.

Researchers from Lanzhou University in China have implemented a novel intelligent point and interval multivariate forecasting system to predict new COVID-19 cases [9]. Their results show improvement on previous forecasting attempts, however, their model is used for prediction of daily cases at the national-level.

On the other hand, researchers from the University of California San Diego have used COVID-19 case number history, demographic characteristics, and social distancing policies to predict the daily trend in the rise or fall of county-level cases [5]. Their findings conclude that the prediction task of county-level case trends is still nontrivial.

We seek to build upon these previous works by tackling the regression problem of predicting daily cases at the county-level.

3 Dataset

The dataset we used was a combination of the following two datasets: a New York Times dataset of county-level COVID-19 cases and deaths [6], and a County Health Rankings and Roadmaps dataset of measurements of county-level health factors [3].

3.1 New York Times Dataset

3.1.1 Introduction

The New York Times dataset contains county-level COVID-19 data for the year 2020[1]. This dataset contains 884,737 observations and 6 features.

Its features include: date, county identification (county, state, FIPS), cumulative case count, and cumulative death count.

A county in this dataset does not appear until it records its first case. Then, it appears every day until the end of 2020.

3.1.2 Feature Engineering

Since we want to predict the daily count of new cases, we subtracted the cumulative case count for a given county and day by the cumulative case count for the

[1] https://github.com/nytimes/covid-19-data

same county one day prior. This new feature represents the case increase for a given county and day. We also subtracted the cumulative death count for a given county and day by the cumulative death count for the same county one day prior. This new feature represents the death increase for a given county and day.

Given that an individual can spread COVID-19 for approximately 14 days after exposure, we created 26 additional features to represent the case and death increases for the previous 13 days.

3.1.3 Pre-Processing

We noticed noise in the dataset; occasionally, the cumulative count of cases or deaths for a given county and date is less than its respective cumulative count for the date prior. In these cases, we dropped the rows containing noise, as well as the next 13 days for those counties (due our feature engineering).

Since we can identify counties by their FIPS code, we dropped the features county and state. Additionally, we also dropped the features cumulative count and cumulative deaths given that the features we extracted make them obsolete.

3.2 County Health Rankings and Roadmaps Dataset

3.2.1 Introduction

The County Health Rankings and Roadmaps dataset contains measurements of county-level health factors for the year 2020[2]. This dataset contains 3,194 observations and 786 features.

Its features include: county identification (eg. county, state, FIPS), health outcomes (eg. premature death), health behaviors (eg. adult obesity), clinical care (eg. uninsured), social and economic factors (eg. unemployment), and physical environment (eg. severe housing problems).

3.2.2 Pre-Processing

Since we can identify counties by their FIPS code, we dropped other county identification features, with the exception of state. We want to keep state as a feature to our models.

We noticed that many of the features have a high number of missing observations. Since we wanted to prioritize the preservation of our observations, we dropped all features with more than 50 missing values. We then dropped all observations that still contained missing values.

The dataset documentation describes that the 'raw value' features are the ratio of the respective 'numerator' features to 'denominator' features. Given this, we dropped the 'numerator' features and the 'denominator' features.

We then performed a one-hot encoding on two categorical features.

3.3 Combined Dataset

After the pre-processing and feature engineering steps, our modified New York Times dataset has 679,629 observations and 29 features. After the pre-processing steps, our modified County Health Rankings and Roadmaps dataset has 3,039 observations and 144. We perform a many-to-one merge of the datasets, merging on FIPS. We then drop FIPS since it is not a feature we want in our analysis. We also drop the death increase for the present day because if we do not have the case increase for that day, it is unlikely that we have access to the corresponding death increase.

The resulting dataset has 649,442 observation and 170 features. For the purposes of our analysis, we reduce our dataset by selecting 5 percent of the observations, chosen at random without replacement. This sampled dataset contains 32,472 observations and 170 features.

4 Problem Formulation

We formulated our problem as a supervised regression problem. We begin by splitting our dataset such that 75 percent is used for training and 25 percent is used for testing. We then separated our label from our features. Our label is the case increase. Our features are the historical case and death increases, as well as its respective county health factors.

We chose to use the Mean Squared Error (MSE) as our loss function. We chose to use MSE because the distribution of our response seems to be exponentially distributed; there are some very large values, but the vast majority are small. Using MSE as our loss function would ensure that our models are paying attention to those extreme values and not ignoring them.

5 Methods

We implemented 5 different models, each with their strengths and weaknesses. We set Ridge regression as our baseline model because Ridge regression is commonly used for similar regression tasks and performs reliably under a large number of situations.

5.1 Ridge Regression

Ridge regression is a standard model for regression tasks. It performs linear regression with an L2 loss and L2 regularization. Ridge regression is well suited for this task because we hypothesize that our problem

[2] https://www.countyhealthrankings.org/explore-health-rankings/rankings-data-documentation

is fairly linear given the available features. Additionally, the L2 regularization does a great job in high-dimensional feature-spaces.

Package: sklearn.linear_model.RidgeCV

5.2 Elastic Net Regression

Elastic net regression is a standard model for regression tasks. It performs linear regression with an L2 loss and a combination of L1 and L2 regularization. Elastic net regression shares the same advantages as Ridge regression, with the additional benefit of performing feature selection via the L1 regularization.

Package: sklearn.linear_model.RidgeCV

5.3 Random Forest

Random Forest is an ensemble method which learns by constructing a multitude of decision trees in parallel. For regression, random forests will return the mean prediction of the individual trees. This corrects for the habits of individual decision trees of overfitting to the training set. We chose to implement a random forest model because of its abilities to learn non-linearities without overfitting.

Package: sklearn.ensemble.RandomForestRegressor

5.4 XGBoost

XGBoost is an algorithm designed by Tianqi Chen that is widely regarded as one of the fastest and most efficient implementations of gradient tree boosting. Gradient tree boosting is a boosting method that learns by constructing a multitude of decision trees in a stagewise fashion. It can be used for regression tasks, and it typically outperforms random forests.

Package: xgboost.XGBRegressor

5.5 Stepwise Regression

Stepwise regression is a methods of training regression models where we systematically decide which features to include. There are a number of selection algorithms for stepwise regression; we implemented forward selection.

In forward selection stepwise regression, we initialize a model with no features. We test the addition of each feature, and add the one which improves the model the most. We repeat this process until there is no significant improvement by adding an additional feature. Our algorithm is outlined:

Algorithm 1 Forward Stepwise Regression

```
Require: $\epsilon > 0$
  $model \leftarrow \emptyset$
  $Err_{old} \leftarrow MAE(model)$
  $\Delta_{Err} \leftarrow Err_{old}$
  while $\Delta_{Err} \geq \epsilon$ do
      try: adding each feature to the model
      Pick the feature, $k$, with the lowest error
      $Err \leftarrow MAE(model + k)$
      if $Err < Err_{old}$ then
          add the feature to the model
          $Err_{old} \leftarrow Err$
      else
          Halt
      end if
  end while
```

For the regression, we used Ridge regression.

Package: sklearn.linear_model.Ridge

6 Experiments and Results

We implemented the models, and provide a description of our hyperparameter tuning, our evaluation metric, each model's performance, and the results of the stepwise regression.

6.1 Hyperparameters

To ensure that our results are reproducible, we outline our hyperparameter selection for each model.

6.1.1 Ridge Regression

We performed 5-fold cross-validation with the following hyperparameter selection:

Alpha: [1, 10, 100, 1000, 10000]

Our best model chose:

Alpha: 10000

6.1.2 Elastic Net Regression

We performed 5-fold cross-validation with the following hyperparameter selection:

L1 Ratio: [.001, .01, .1, .5, .9, .99, 1]
Alpha: [1, 100, 250, 500, 1000, 2000, 4000]

Our best model chose:

L1 Ratio: [.001, .01, .1, .5, .9, .99, 1]
Alpha: [1, 100, 250, 500, 1000, 2000, 4000]

6.1.3 Random Forest

Given the research proving that random forests do not benefit much from hyperparameter tuning [7] [8], we decided to forgo hyperparameter tuning, and instead, chose the following recommended values

Number of Estimators: 1000
Max Features: $\sqrt{p}$

6.1.4 XGBoost

We performed searched for the optimal hyperparameters with the following hyperparameter selection:

Number of Estimators = [100, 200, 500]
Learning Rate = [.001, .01, .1]
Max Depth = [5, 7, 9]

Our best model chose:

Number of Estimators = 200
Learning Rate = 0.01
Max Depth = 9

6.1.5 Stepwise Regression

We did not perform a hyperparameter search for the stepwise regression because we wanted to ensure that our model would perform enough feature selection. If we were to search for an optimal value of alpha (like in Ridge regression), our model would tend to choose a large value of alpha and include lots of features instead of performing feature selection. Our model uses:

Alpha: 1
Tolerance: 0.01

Where tolerance is the minimum difference in error required to include a feature.

6.2 Evaluation

We evaluated our models using the Mean Absolute Error (MAE) metric. We chose this metric over the popular Mean Squared Error (MSE) because we cannot control for the population of a county (which affects our residuals), and so we want our penalization to be linear with the residuals.

Note that we intentionally use the MSE as our loss function and the MAE as our evaluation metric. This is because we do not want our models to ignore extreme values when training, but when we test our models we want the penalization to be linear.

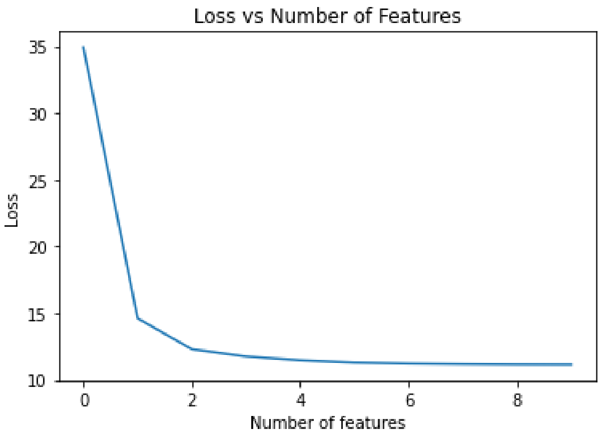
6.3 Performance

We report the following model performances:

Model	Mean Absolute Error
Ridge Regression	11.90
Elastic Net	11.80
Random Forest	12.04
XGBoost	11.38
Stepwise Regression	11.33

6.4 Feature Selection

We notice the change in loss as we add more features.



We also observe that the following features were selected by the stepwise regression, in this order:

1. Case increase 7 days ago
2. Case increase 1 days ago
3. Case increase 6 days ago
4. Case increase 5 days ago
5. Population raw value
6. Case increase 2 days ago
7. Case increase 4 days ago
8. Percent rural raw value
9. Case increase 3 days ago

7 Conclusion and Discussion

We will now summarize our findings, mention major takeaways, propose directions of future research, and reflect on what we learned.

7.1 Summary of Findings

We begin by noting that stepwise regression and XGBoost shared relatively similar performances and that the random forest was the worst performing model. We also note that stepwise regression had an alpha value of 1, whereas Ridge regression and elastic net regression used alpha values of 10000 and 2000, respectively.

We hypothesize that the reason for these results is that only a small amount of the features are significant.

Stepwise regression does a great job of feature selection. It begins with an empty model, and iteratively adds the feature that is the most significant (up to a certain threshold). Our model only selected 9 of the original 169 features.

XGBoost also does a good job of feature selection. The individual trees are constructed in a stagewise fashion, and each tree has a maximum depth of 9. Therefore, XGBoost only select the features that are most significant.

Elastic net attempts to perform feature selection. The L1 regularization zeroes out some features, and in the end, elastic net had 19 out of 169 non-zero coefficients.

Ridge regression is unable to perform feature selection. All 169 coefficients are non-zero. Given that the model has 169 non-zero coefficients, and most of them are not significant, it attempts to regularize by picking a very large value of alpha.

Our random forest algorithm struggles with this dataset. Since each of its individual tree are constructed in parallel, we hypothesize that it is unable to learn which features are significant and which ones are not.

7.2 Takeaways

We conclude that it is in fact possible to predict daily COVID-19 cases using a county's historical case and death increases, as well as its county health factors.

The successes of our stepwise regression and XGBoost models compared to the results of our random forest and Ridge regression seem to imply that only a small subset of our features are actually significant when predicting new COVID-19 case counts. While it is not entirely surprising that only some of the features are significant, it is interesting to note which ones were chosen in the forward selection stepwise regression.

The stepwise regression algorithm selected historical case increases from the past 7 days. This makes sense because people are most contagious during the 7 days after COVID-19 exposure. It is interesting to note, however, that the algorithm did not select any features corresponding to the historical increases in death counts.

From the wide selection of county health factors, it is interesting to note that the algorithm only selected 2 features: population raw value and percent rural raw value. We believe that the algorithm selected percent rural raw value because of its correlation with population density. In that case, we find it interesting that the algorithm deemed features relating to population to be most significant of increases in COVID-19 cases.

7.3 Future Directions

While we were able to make relatively accurate predictions of daily COVID-19 case counts, there is much work to do. The biggest area of improvement that we can identify would be to incorporate a count of active cases and to try to predict the percent change.

We hypothesize that incorporating a count of active cases and predicting the percent change would resolve a large number of our challenges. Our response variable, new cases, varies from 0 to 30,000, has a mean of 25, and a median of 3. It is incredibly difficult to predict the count of new cases when its distribution appears to be exponential.

Incorporating a count of active cases and attempting to predict the percent change would dramatically reduce the range of possible values. Additionally, we hypothesize that the percent change would approximately follow a normal distribution, which would further help with the predictive abilities of the model.

7.4 Learnings

By completing this final project, we were able to gain a lot of knowledge and experience that we otherwise would not have gained. The biggest lesson we learned is the importance of thinking and reasoning through a machine learning problem. If we want to predict a target variable from a set of features, we must first analyze the problem and ask ourselves if machine learning can tackle this problem. Generally speaking, machine learning is a combination of probability, linear algebra, and computation, and so it is very important to make sure the problem can be solved using some combination of the three.

Acknowledgments

We would like to thank Professor Lyle Ungar and our Teaching Assistant Parth Sheth for their continued advice and guidance throughout this project. Additionally, we would like to thank all the CIS 5200 staff for an amazing semester. We learned a lot and are very thankful for your support.

References

[1] American Journal for Managed Care. A Timeline of COVID-19 Developments in 2020. https://www.ajmc.com/view/a-timeline-of-covid19-developments-in-2020.

[2] Centers for Disease Control and Prevention. About COVID-19. https://www.cdc.gov/coronavirus/2019-ncov/your-health/about-covid-19.html.

[3] County Health Rankings & Roadmaps. National Data & Documentation: 2010-2020. https://www.countyhealthrankings.org/explore-health-rankings/rankings-data-documentation/national-data-documentation-2010-2019.

[4] Johns Hopkins Bloomberg School of Public Health. COVID-19 Testing: Understanding the âPercent Positiveâ. https://publichealth.jhu.edu/2020/covid-19-testing-understanding-the-percent-positive.

[5] Megan Mun Li, Anh Pham, and Tsung-Ting Kuo. Predicting covid-19 county-level case number trend by combining demographic characteristics and social distancing policies. *JAMIA Open*, 2022.

[6] New York Times. Coronavirus (Covid-19) Data in the United States. https://github.com/nytimes/covid-19-data.

[7] Philipp Probst and Anne-Laure Boulesteix. To tune or not to tune the number of trees in random forest? 2017.

[8] Philipp Probst, Marvin N. Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3), jan 2019.

[9] Zongxi Qu, Yongzhong Sha, Qian Xu, and Yutong Li. Forecasting new covid-19 cases and deaths based on an intelligent point and interval system coupled with environmental variables. *Frontiers in Ecology and Evolution*, 10, 2022.

[10] Del Guercio K et al. Ritchey MD, Rosenblum HG. Covid-19 self-test data: Challenges and opportunities â united states, october 31, 2021âjune 11, 2022. 2022.