A Macro-Metric Machine Learning Prediction for CoronaVirus Propagation

Jacob Evarts
Department of Computer Science
University of Oregon
jevarts@uoregon.edu

Nolan Rudolph
Department of Computer Science
University of Oregon
ngr@uoregon.edu

ABSTRACT

The current unprecedented spread of COVID-19 has affected almost all aspects of life around the world. With resources spread thin and lives on the line, any additional insight into the state of global health can be vital. In this project, we've applied ensemble learning in the form of gradient boosting regressions and propose that our solution will aid in understanding the spread of an elusive disease in an environment without adequate testing supplies.

1 INTRODUCTION

Whether you have intentionally searched for it or not, any social media platform or forum page will have multiple threads regarding the notorious CoronaVirus (COVID-19). Empirically, the virus has increased in severity due to the invoked concern of the global population. Statistically, the spread of COVID-19 exhibits exponential growth given the data of confirmed infected individuals from a multitude of countries, supporting the terrified consensus of the general public. With the rapid proliferation of COVID-19 creating an international crisis, we have found it in the public's best interest to develop a machine learning program capable of diagnosing the future pandemic status of the CoronaVirus to assist in mitigating its spread.

We are very fortunate to have various datasets provided to us by John Hopkins University [1], holding CSV formatted data that provides insights on confirmed infected cases. In these datasets, each case is documented with a range of attributes, where the ones we intend to emphasize are:

1. Reported date
2. Region
3. Number of confirmed cases
4. Number of caused deaths
5. Number of recovered cases

All attributes apart from the *reported date* have been selected as most important to our project due to the macro-metric nature of our program. The reported dates stretch as far back as January 22th, 2020, and the most recent cases are constantly appended to these datasets, providing an encasement of the entire epidemic.

The question then remains, what is the particular objective of our machine

learning program? Trivially, our program will be able to learn from our massive training dataset and then be able to accurately predict some secluded test subset, unseen to the machine or human eye. This is the expected result, however it provides very little insight into how the disease will behave and spread in the future.

Therefore, we propose that our machine learning algorithm will be able to take on new datasets of fabricated information regarding individuals from all over the globe and decide the probability that such individuals will become infected. Since epidemics exhibit a fractal pattern when populations are divided into communities, i.e. land/border divided countries, each community functions as if it were a singularity [2] due to the initial travel of infected individuals. It is for this reason that we are able to use figurative situations to diagnose where the future spread of the CoronaVirus will be, and the likelihood of such an event. This is the ultimate goal of our machine learning project.

2 BACKGROUND

With the objective clearly stated, we turn our attention to the logistics of our program. Considering a single entry from our training dataset, we would like to predict whether or not the individual will become infected. This translates to a logical outcome or a binary classification of any given entry. Any uncertainty should favor the side whose endpoint is closer to the estimated value (i.e. a score of 0.49 in a binary classifier would become a classification of 0 whereas 0.51 would become 1). Therefore, we should only contemplate training models who utilize a labeling method of this caliber.

Another important subject of consideration is how accurate we need our model to be. As our project focuses on the macro-metric aspects of the CoronaVirus, concerning itself with vast regions and their communities, we must ensure that our machine learning model is as precise as computationally possible. In a realistic sense, misclassification of a region's likelihood of infection could lead to the extraneous efforts of millions of individuals, or in the worst case, the infecting and potential death of catastrophic numbers. Therefore, whereas some machine learning models may be able to reliably and efficiently achieve an accuracy of 90%, we must strive to find a model that emphasizes precision near 100%. Although the ideal model would achieve satisfactory precision and efficiency, the fundamentals of non-deterministic polynomial-time in regards to machine learning training cause an inversely proportional relationship between efficiency and precision [3]. Therefore, we must heavily favor machine learning models that achieve precision over efficiency.

3 METHODS

First and foremost, we needed a method of importing our datasets into our modules to initialize the preprocessing of the data. We did so by using a data analysis library named Pandas [4], where we inserted,

deleted, and renamed different columns of the dataset to our liking. We then used SciKit-Learn's (SK-Learn) Label Encoder [5] and One Hot Encoder [6] to allow our dataset to conform to the expectations of our model. This allowed each region to have its own column where a boolean value denotes whether or not a specific entry includes a respective region.

We decided to use SK-Learn's prebuilt machine learning models [7] in an effort to avoid gratuitous debugging and extraneous coding. However, we were faced with deciding what model would be most practical for our project. Granted our emphasis on the importance of precision, we began by conducting a cross-validation of various models entailed by the SK-Learn repository, where we came by a model that perfectly suited our needs.

This model was the Gradient Boosting Regressor, an ensemble learning technique that capitalizes on fixing its own mistakes. In each successive model of its training, it uses its previous model to adjust its weights depending on whether or not a classification was correct. Letting φ be the learning rate, w' be the previous model's weights, and $base = sum(w' * e^{(-1 * \varphi)}) + sum(w' * e^{\varphi})$, then each correct prediction receives an updated weight of $(w' * e^{(-1 * \varphi)}) / base$, and each incorrect prediction receives an updated weight of $(w' * e^{\varphi}) / base$. Here we witness how incorrect predictions have a much greater influence on the weight of the model.

In this sense, the gradient boosting regressor acts as its own cross-validation system, which flawlessly appeases our requirement of near 100% precision. After training our model for approximately eight hours, we were able to configure the hyperparameters of our model in a way that optimized our predictions. Hyperparameter tuning was performed using SK-Learn's GridSearchCV [9]. We then performed exhaustive testing of parameter values for the model, including the number of estimators, the max depth of each estimator, and the learning rate. Using a mean squared error function to quantify our accuracy with respect to linear regression, we found that our model was able to reduce itself to an error rate of less than a 10th of our baseline functions, an outcome that blew our expectations out of the water.

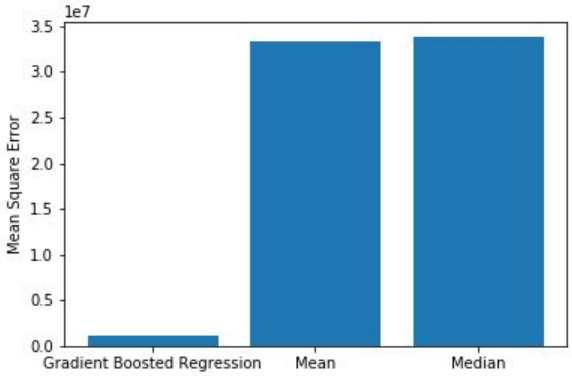

Figure 1: Gradient Boosting Mean Squared Error with respect to Mean and Median

We had undoubtedly chosen the correct machine learning model, and it was time to begin evaluating its performance on integral datasets.

4 CONCLUSION

We found that machine learning models, particularly gradient boosting regressions, can yield promising results in regards to the notoriously unpredictable spread of diseases. Our R-squared value for the model was 0.951, so the features chosen to model the number of confirmed cases were excellent predictors of the actual values. Additionally, the test error was 13.5% higher than our validation error, suggesting that overfitting was not a large problem, as could be expected when using ensemble learning.

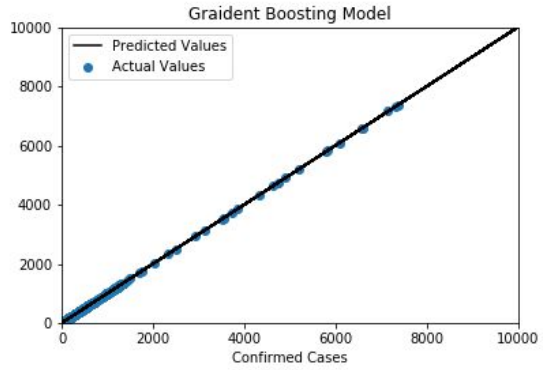

Figure 2: Gradient Boosting Model Prediction with respect to Actual Labels

Given the large number of factors that affect the spread of disease, particularly at a macro-level scale, it is a challenging inference question. The various public policies in each prediction region could be extremely insightful to include, as well as the amount of funds, hospitals, and staff each region has dedicated to the pandemic. This model is proof that spending additional resources on data collection and model architecture would be valuable to understand what to expect as elusive conditions like the COVID-19 spread.

5 REFERENCES

[1] "Novel CoronaVirus 2019 Dataset," *Kaggle*. Accessed: 09-Mar-2020. [Online]. Available: https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset.

[2] "Fractal Structures and Fractal Functions as Disease Indicators," *ResearchGate*. Accessed: 11-Mar-2020. [Online]. Available: https://www.researchgate.net/publication/272531043_Fractal_Structures_and_Fractal_Functions_as_Disease_Indicators.

[3] Guo, Haipeng & Hsu, William. "A machine learning approach to algorithm selection for NP-hard optimization problems: A case study on the MPE problem. Annals of Operations Research.". *ResearchGate*. Accessed: 12-Mar-2020. [Online]. Available: https://www.researchgate.net/publication/225149774_A_machine_learning_approach_to_algorithm_selection_for_NP-hard_optimization_problems_A_case_study_on_the_MPE_problem

[4] "Python Data Analysis Library." Accessed: 09-Mar-2020. [Online]. Available: https://pandas.pydata.org/.

[5] "Scikit-Learn Preprocessing LabelEncoder," *scikit-learn*. Accessed: 10-Mar-2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[6] "Scikit-Learn Preprocessing OneHotEncoder," *scikit-learn*. Accessed: 10-Mar-2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html#sklearn.preprocessing.OneHotEncoder

[7] "Scikit-learn, Machine Learning in Python," *scikit-learn*. Accessed: 09-Mar-2020. [Online]. Available: https://scikit-learn.org/stable/

[8] Nolan Rudolph, Jacob Evarts, A Macro-Metric Machine Learning Prediction for CoronaVirus Propagation, GitHub repository, https://github.com/NolanRudolph/CornaVirusMLPrediction

[9] “Scikit-Learn GridSearchCV,” *scikit-learn*. Accessed 08-Mar-2020. [Online]. Available: https://scikit-learn.org/stable/