Regression for Modeling Covid-19 Data (*Machine Learning Final Project*)

Sommer Harris
harris.som@northeastern.edu

Mitchell Lawson
lawson.mi@northeastern.edu

David Maunder
maunder.j@northeastern.edu

Ian Yip
yip.i@northeastern.edu

Abstract

This paper reviews the role of machine learning in disease modeling, specifically in the context of Covid-19. It reviews papers that combine machine learning techniques with Susceptible Infected Removed (SIR) disease modeling. We then explain our own model, using linear regression to predict new Covid-19 cases, using the 'Our World in Data' dataset. Using R^2 and Root Mean Squared Error, we compare Principal Component Regression (PCR) and Partial Least Squares (PLS) regression and find that at a lower number of components, PLS performs better. For both in different testing scenarios, as we increase components our results worsened. Our regression model did not perform as well as current state of the art models, but did demonstrate that there is potential in using linear regression for predicting Covid-19 cases.

1. Introduction

1.1. Motivation

The paper proposes using machine learning to create a regression model for Covid-19. In a world afflicted by a global pandemic, with a large population of individuals greatly susceptible to this virus, technology and concepts learned from machine learning can be a useful tool in aiding public health officials and the general population in the right direction for mitigating the pandemic's effects. By modelling and predicting Covid-19 data, we can better understand the factors that increase the spread of Covid-19, how to prevent further spread and promote awareness. One of the most novel applications of such a model is aiding government and health care workers to make ideal decisions despite there being areas where people have lower access to the equipment, technology, education and/or experience in the area of machine learning or virus transmission and epidemiology. Another area of motivation that we have identified is to truly determine the relationship and/or effect that immunization plays in stunting the spread of this virus. Furthermore, learning more about what machine learning models and techniques prove to be useful in this field allows us to be better prepared for future pandemics and even be used to save lives.

Our individual learning goals that align with this project are to learn about regression, as well as the ability to train computers to pick out insights in data even when the programmer is not an expert in that field. We hope to do this through developing a linear regression model to predict Covid-19 cases.

2. Related Works

2.1. Primary Research

In our preliminary review, we explored both simple regression models, as well as more complex data models that use differential equations for the SIR(Susceptible, Infected, Recovered) format. First we will describe the regression model we plan to replicate, then we will share our research on how the SIR formulation also could be used to represent our data.

In one paper we studied, Gupta, et. al modeled the Covid-19 spread pattern using a generalized regression neural network, enhanced by a flower pollinator algorithm (FPA-GRNN) [3]. This paper compared the performance of three network models, including the aforementioned one, a non-linear regression, and a support vector machine. The root mean squared error was compared across these algorithms. The accuracy of the FPA-GRNN was better than the other two algorithms.

A second paper that used regression to model the spread of Covid-19 proposed partial derivative regression and non-linear machine learning to improve their model, outperforming current state of the art models [8].

Lam Harrison's GitHub "UK Coronavirus (COVID-19) Machine Learning Prediction" demonstrates a similar approach, using regression to model the Covid-19 trend. This model uses transfer learning, applying a multilayer perceptron. The data is incredibly simple, just two columns of numbers (days and cases) for each country data [6]. This

was the first dataset that we explored working with before deciding it was too simple for our model.

Before we narrowed the scope on this project to a regression, we looked at how researchers model disease with the SIR model. Machine learning and SIR model can be combined to track changes in government policies, and include them as forecasting parameters [13]. Alanazi, et al. also used the SIR model to inspire projections accounting for several possible government actions: "no actions", "lockdown" and "new medicines" [1]. Some authors, such as Pinter, et al. examine machine learning approaches such as adaptive network-based fuzzy inference system (ANFIS) and multi-layered perceptron-imperialist competitive algorithm (MLP-ICA) as alternatives to the SIR model [11]. Farooq, et al. created an intelligent model where parameters are continually updated with new data using ANN based incremental learning approach [2], integrating both the SIR model and non-linear regression. Authors such as Roberts suggests that the best way to accurately model Covid-19 is to conglomerate a combination the models available to us [12]. This last paper may inform how we can think both about the role of the SIR model and a simple regression in epidemiological modeling.

While our model was ultimately much simpler than the examples mentioned above, this preliminary research did show us the broad landscape of this conversation, as well as help us find the dataset we ended up using.

2.2. Evaluation Methods

Evaluation plays a role both in judging our final model, as well as the validation process where we adjust our parameters to make the best model possible. Because our parameters are highly dependent on the data we use, our data selection, mentioned in the following section, is crucial to the evaluation process.

Our regression model uses case counts as the target we are aiming to predict. Predicting death rates was another target we explored, but we chose case counts over death rates because there was more information on case counts in the dataset.

There are many different measures used to attest to the correctness or accuracy of a certain regression model, some of which are square error, mean squared error, root mean squared error, relative mean squared error, the coefficient of determination (the correlation coefficient squared), absolute error and lastly mean absolute error. After reviewing the metrics from other studies we decided it was best to use the metric of root mean squared error, which we examine first; we also explore R^2 as a metric.

We needed to design our process for splitting the data and training. One method that was used by Alanazi et al. was training a newly generated model for one country's data then applying it to another country to determine how accurate it performed. In this paper they examined a model fitted to Italy and then applied it to China to determine how well it predicted actual Covid-19 rates. We thought that we may have been able to improve upon their technique of validation and testing, and reduce overfitting, by using countries that share more similarities [1]. However, our model produced poor results when attempting transfer learning, so we decided just to focus on improving our model with UK data. An additional method of testing our evaluation of the model is to compare with existing models to have another baseline to compare against (did our model predict what occurred more accurately). As we refine our model, we can use cross validation to tune hyper-parameters, and assess whether regularization of the parameters improves our model. We can display and assess this information with feature importance plots.

Another revision we will make while evaluating our model is to see how dimensionality reduction (principal component analysis) may aid in predictions as there perhaps could be confounding variables and/or parameters.

Our objective is to build an initial regression model as a baseline, using the dataset from Our World in Data [4]. We can use squared error to asses and individual model and then compare our score with that of other models (or different parameter configurations of our model). Success would be improving upon our base model, learning more information about what approach works for this sort of modeling, or determining which parameters from Our World in Data play the biggest role, and correctly weighting them.

We used the following cost models for root mean squared error (RMSE) and R^2:

RMSE:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{N}}$$

where:

$$N = number\ of\ elements$$

$$\hat{y_i} = predicted\ amount$$

$$y_i = actual\ amount$$

R^2:

$$R^2 = 1 - \frac{sum\ squared\ regression (SSR)}{total\ sum\ of\ squares\ (SST)}$$

$$= 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y})^2}$$

$$y_i = actual\ amount$$

$$\hat{y_i} = predicted\ amount$$

$$\bar{y_i} = mean\ amount$$

The reasoning behind using R^2 or the coefficient of determination as a metric is that it is representative of the proportion of variation in an outcome or target that is explained by our predictor variables or features such as handwashing facilities, smokers, vaccinations, etc. In general, the higher the R^2 value is, the better the model should perform, but correlation does not necessarily equal causation.

Similarly, root-mean-squared-error is another metric used in the analysis of our models because it is the average error performed by the model in predicting the outcome for a given observation. Depending on the scale of the values of our features, this can affect the value of the MSE but not necessarily the relative performance of the model to predict the target value. Because we did not scale values, we decided to use RMSE as a criterion for finding the best model, as it is not affected by whether or not the target variables are scaled and allows us to do less preliminary transformations on our data before inputting it into the model.

2.3. Resources

Initially, we started with a simple dataset such as the one on Lam Harrison's GitHub [6]. However, as we considered incorporating more parameters to build the model, it is more realistic that we will pull from a dataset such as that of Our World In Data. This dataset is a comprehensive resource, contributed to by prominent organizations such as Johns Hopkins University, Oford, the UN, and the world bank, among others. On their github, the editors outlines metrics from vaccinations, tests, and confirmed cases, to hospital information, and virus reproductive rate and stringency index (government response) [4].

For defining parameters such as reproductive rate (informed by government lockdown policy, social distancing, and mask wearing) we began using estimates from the "Our World in Data" Covid-19 dataset [4].

The code was written in Python, and was in a .py file. The main file used for plotting was pcr.py. Modules that will be used include sci-kit learn [10], numpy [5], pandas [9] and matplotlib [7]. Sci-kit learn is a python package with pre-created models for machine learning. Pandas is a datascience library that allows us to work with dataframes more efficiently. Numpy is another package that allows us to perform matrix multiplications efficiently. Additionally, sci-kit learn is built on numpy, and numpy arrays can be inputted into sci-kit learn models. Matplotlib is the package that will be used to plot results. All code was run using our laptops.

3. Proposed Method

Our project relied heavily on a trial and error processes that helped us determine which elements were best suited to our model. We first attempted to hand pick the features that would be used in our regression model. However, this

```
Index(['iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases',
       'new_cases_smoothed', 'total_deaths', 'new_deaths',
       'new_deaths_smoothed', 'total_cases_per_million',
       'new_cases_per_million', 'new_cases_smoothed_per_million',
       'total_deaths_per_million', 'new_deaths_per_million',
       'new_deaths_smoothed_per_million', 'reproduction_rate', 'icu_patients',
       'icu_patients_per_million', 'hosp_patients',
       'hosp_patients_per_million', 'weekly_icu_admissions',
       'weekly_icu_admissions_per_million', 'weekly_hosp_admissions',
       'weekly_hosp_admissions_per_million', 'new_tests', 'total_tests',
       'total_tests_per_thousand', 'new_tests_per_thousand',
       'new_tests_smoothed', 'new_tests_smoothed_per_thousand',
       'positive_rate', 'tests_per_case', 'tests_units', 'total_vaccinations',
       'people_vaccinated', 'people_fully_vaccinated', 'total_boosters',
       'new_vaccinations', 'new_vaccinations_smoothed',
       'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
       'people_fully_vaccinated_per_hundred', 'total_boosters_per_hundred',
       'new_vaccinations_smoothed_per_million', 'stringency_index',
       'population', 'population_density', 'median_age', 'aged_65_older',
       'aged_70_older', 'gdp_per_capita', 'extreme_poverty',
       'cardiovasc_death_rate', 'diabetes_prevalence', 'female_smokers',
       'male_smokers', 'handwashing_facilities', 'hospital_beds_per_thousand',
       'life_expectancy', 'human_development_index',
       'excess_mortality_cumulative_absolute', 'excess_mortality_cumulative',
       'excess_mortality', 'excess_mortality_cumulative_per_million'],
      dtype='object')
```

Figure 1. Column names used in Our World in Data's Covid-19 dataset. Our parameters are a subset of the columns listed here, and we reduce the number through Principal Component Analysis

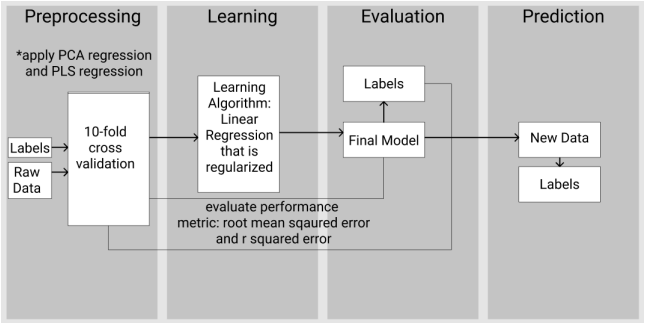

Figure 2. Our Supervised Learning Process

method can be prone to error and personal bias. Instead, we decided to run a form of dimensionality reduction along with our regression model.

As outlined in Figure 2, above, we followed the general flow of supervised learning: pre-processing, learning, evaluation, and prediction.

In the pre-processing phase we applied two dimensionality reduction methods: Partial Component Regression(PCR) and Partial Least Squares (PLS) Regression. For both PCR and PLS, we split the data in the pre-processing stage using 10-fold cross validation to evaluate the performance of the model. We simulated the predictive power of our model by using the shift function in pandas. This allowed us the shift the columns of the dataframe up. We would then attempt to predict how many cases were in a day with data such as new cases, new cases smoothed, ect. from the previous day or previous month.

PCR applies Principal component analysis (PCA) to the data first before running a linear regression model on the data. Additionally, we also scaled the data because PCA reduces dimensions by finding ones where the variance is highest. Some dimensions have different scales, which can affect the dimensions that PCA chooses. We created a pipeline to combine these steps. This pipeline allowed us to customize each step and allowed it to work better with other Sklearn packages like the one used for cross validation. The linear model chosen was L2/Ridge regression.

PLS on the other hand does the dimensionality reduction and regression in a single function. An advantage of PLS is that it takes into account the output when doing the regression. When there is low variance in the correlation direction, PLS should perform better than PCR because PCA only cares about the variance and not the predictive power of a dimension.

PCR:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h(x_i) - y_i)^2 + \lambda\sum_{i=1}^{n}\beta_n^2$$

$$J(\theta) = \frac{1}{2}(x\theta - y)^T(x\theta - y) + \lambda\sum_{i=1}^{n}\beta_n^2$$

PLS:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h(x_i) - y_i)^2$$

$$J(\theta) = \frac{1}{2}(x\theta - y)^T(x\theta - y)$$

$$\theta = \begin{bmatrix}\theta_1\\ \theta_2\\ \vdots\\ \theta_n\end{bmatrix}$$

$$x = \begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}$$

$$y_i = actual\ amount$$

The following is a formalized mathematical representation of a linear regression model. We used this approach to predict the target value, number of Covid-19 cases.

$$h(x) = \sum_{i=1}^{n}\theta_i x_i + \beta$$

Beta is the hypothesis when all independent variables are zero.

$$\theta = \begin{bmatrix}\theta_1\\ \theta_2\\ \vdots\\ \theta_n\end{bmatrix}$$

$$x = \begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}$$

4. Experiments

Our dataset was the "Our World in Data" Covid-19 dataset [4]. This dataset has time series data about the number of Covid-19 cases around the world. It also has important variables that could affect Covid-19 numbers such as the number of tests done, a rolling average of cases and a stringency index measuring government measures. Figure 1 shows the columns of our dataset. Specifically, we focused on the regressing to predicting the number of cases happening in the UK on a certain day. We chose the UK because the data was cleaner than other countries. The number of new cases did not reach extreme negative values and the columns in the dataset were mostly filled. For example, in the dataset, we tried completing a global regression model, but found that Spain once reported a new case count of 70,000 in a day. In our dataset, there were 643 rows of UK data.

After dropping all of the countries other than the UK, we filled null values with 0. A lot of null values were early in our dataset, where new cases and new cases smoothed would be not filled in. This suggested that they should be equal to 0. Next, date was changed to an integer because Sci-kit learn cannot run a regression on a date format. We formatted the date columns as a date time and then changed that to an ordinal number. Finally, all non-numerical data was dropped because the regression cannot run on text/categorical variables.

We then created a copy of the number of new cases into an output variable. Next, we shifted the data by the number of days we wanted to predict in the future. For example, if we were trying to predict 1 day into the future, we would have all of the columns from the previous day. Several columns in the data were also moving average data for the week, such as the 7 day average of cases and tests done and thus provided the model the previous week's average cases. For example, by shifting 30 days, if we were predicting the number of cases on December 14, we would give our model the column values on November 14 which included the average number of cases, tests and other values from the previous week down to November 7.

Finally, we ran both PLS and PCR on the data from one component to fourty components. We took the aver-

age RMSE and R squared of the 10 fold cross validation and recorded that as our results. We ran a ten fold cross validation on 643 rows corresponding to 643 days of UK Covid-19 data. We did cross-validation since it would give us the most unbiased results and would not be affected by a random seed of a train test split.

5. Results and Discussion

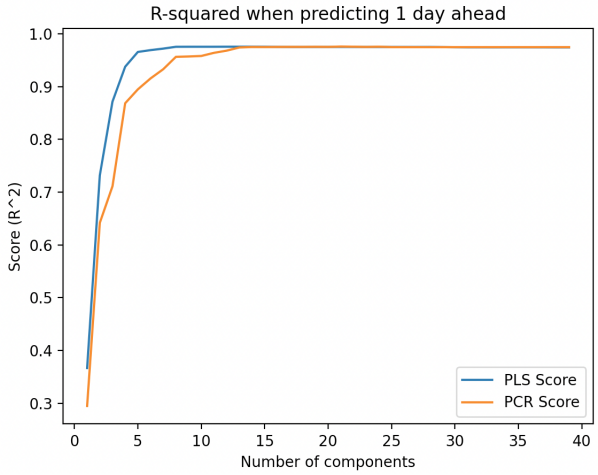

Figure 3. R squared when predicting 1 day ahead

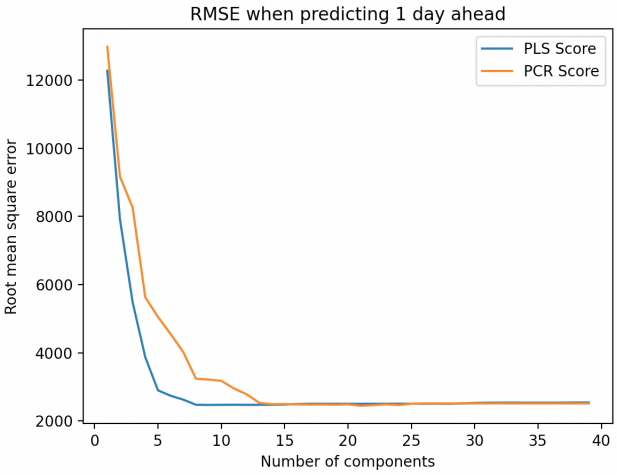

Figure 4. RMSE when predicting 1 day ahead

At first the group had tried training a model using all of the global data but this presented many issues when using the model to predict values for an individual country. The results did not show any accuracy and had high error.

Although this is real world data with many sources collection presenting a high degree of error in the collection process, we acknowledged some of these sources and tried to handle them accordingly. We did so by narrowing our

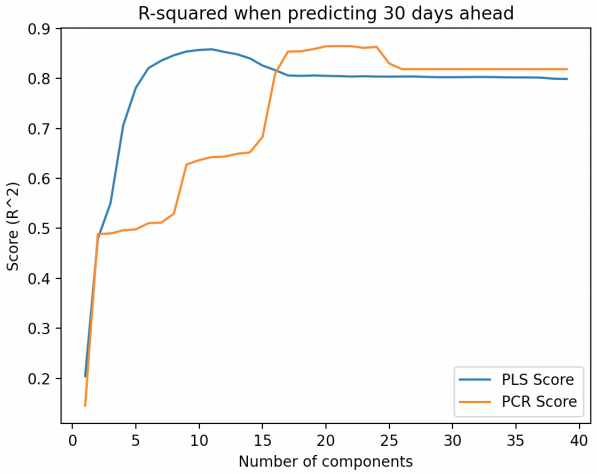

Figure 5. R squared when predicting 30 days ahead

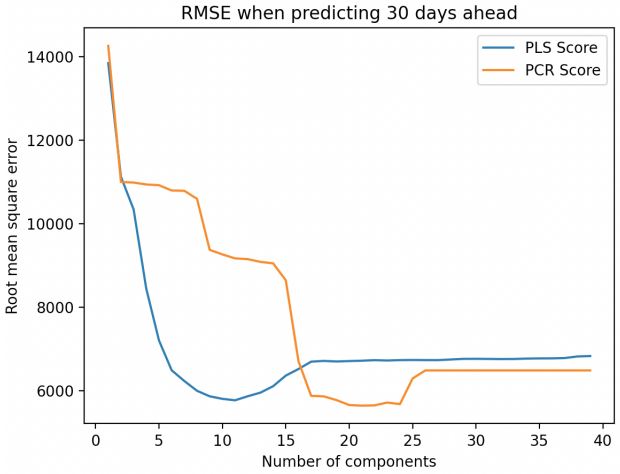

Figure 6. RMSE when predicting 30 days ahead

dataset, focusing on quantity over quality, investigating 2 different types of models and dropping clearly unnecessary or confounding columns in the data. The latter point meant excluding the columns 'new cases smoothed' and 'new cases smoothed per million' from our analysis, as these columns contain some of the actual target values, and we didn't want to leak them into our model.

The criteria that we used to judge the relative performance of our model and the results we obtained included R^2, the coefficient of determination and root-mean-squared-error (RMSE). The plots produced after training our model are based on data from the UK, and are presented in Figures 3-6 where there are 2 prediction periods each trained with 2 different models, resulting in a key differences that are evident in these plots.

Firstly, our RMSE is very high in both Figure 4 and Figure 6, the 1 day and 30 day ahead models respectively. The

minimization of RMSE appears to be about 10 components in the case of PLS and about 15 in the case of PCR, with these approximations being consistent in both of the aforementioned figures.

For R^2 we can see that both the PLS and PCR in Figures 3 and 5, the 1 day ahead and 30 days models, exhibit their optimal scores for the same number of components as the optimal values for the RMSE figures. This is interesting because it can be observed that these graphs almost appear to be inverses or rather a reflection of one another.

In general for RMSE and R^2, PLS appears to have a smoother trend. Conversely, the 30-day ahead models have a more clear and defined optimum than the 1-day ahead models.

Now this does not come as a surprise, but it's obvious that the 1-day ahead model performs better than the 30-day ahead model because practically speaking it's easier to predict 1 day than it is to predict 30 days ahead. There is more data closer to the points that we are trying to predict, resulting in a better prediction.

Although the PLS and PCR models appear to perform similarly, we focused on the PLS model because it appeared to perform better earlier in our project as well as our results, it is able to reach the best MSE and R^2 value using a lower number of components, in addition to accounting for the variation in the target variable as well as the features, where PCR only accounts for the variation in the features.

It's unfortunate that the RMSE from our models in Figures 4 and 6 show high error, but it may be due to the fact that Covid-19 data has such high variability in it, which is a point that we hope to identify and alleviate in future work.

6. Conclusions

6.1. Future Work

To make our model more accurate and verbose we can focus on improving our model by making five changes:

1. Add more detailed parameters to our model
2. Handle outliers in a more precise fashion
3. Change our model to a multivariate model.
4. Make a more robust worldwide model
5. Converting our Model to a SIR Model

By adding more precise and detailed parameters to our model we can better predict Covid-19 cases on a day-to-day basis for each country. Our model includes various important independent variables that are strong predictors of future cases such as previous daily cases and amount of people who are vaccinated but does not include certain qualitative factors that can significantly affect daily cases. Two of these qualitative factors would be the emergence of a new variant and weather factors (rain, cold, or extreme heat). Since these factors cannot be easily pulled from the internet in a quantitative factor, we would have converted these qualitative components into a quantitative format. For example, we could assign a value to how effective the disease is as spreading based on scientific research done by the World Health Organization. We would then plug this new quantitative factor into our model as a parameter.

The second improvement that could help benefit our model further is through the handling of outliers in order to effectively and efficiently lower the variance in our data which would have a direct and positive impact in improving RMSE. In our current implementation of our models, we do not intervene or handle outliers in any meaningful way. Given more time to investigate and test further would mean us being able to explore 2 novel avenues for handling these outliers. Firstly, a possible way of identifying outliers is by finding the squared error on all our samples, then sorting them such that we produce a graph that allows us to look for discontinuities. If there is a sudden spike that creates a high plateau or event we would be able to identify outliers. Secondly, an alternative and possibly even more clever way we could identify outliers in the owid Covid-19 data that would be effective with a large amount of features is by using a binary classification algorithm like logistic regression to help us identify these outliers without us having to explicitly identify examining them directly. In either solution, we could then choose to handle outliers by possibly excluding them altogether, study them further to try and gain some understanding of what causes them, or even replace them with the mean of our data or 0.

In the future we can attempt to improve our model by adjusting the outliers by converting them to the average Covid-19 cases. This could potentially improve our model in two ways. The first way would be to add more Covid-19 cases to the first five months of Covid-19. This potentially could improve our model, since our model is including small amount of daily Covid-19 cases when the government was unsure about the severity and spread of Covid-19 (Jan 2020-March 2020) due to lack of Covid-19 testing kits. This approach could also improve our model by allowing us to include countries with negative daily Covid-19 cases. Countries like Spain and Portugal had negative or zero daily Covid-19 cases during high daily Covid-19 times. This was due to fixing errors or batch Covid-19 reporting (reporting three days in one). By averaging these amounts out, we could include these countries into a worldwide Covid-19 model.

Daily Covid-19 cases are very important to predict as this is the basis for hospitalization and ICU count, but as vaccinations and other medication become more common, having our regression model predict other factors like deaths,

ICU count and hospitalizations could significantly help the medical field make more accurate decisions. This means would just have to convert our model from a linear regression model to a multivariate linear regression model. In the future we could improve our model by creating a global model. By aggregating our first three changes we could compare how single country models perform against global models in terms of predicting daily Covid-19 cases. The last change we would make to our model in the future would be to convert our model into a SIR Model. We would make this change by adding our linear regression coefficients into a SIR model for each component of the SIR Model. By doing this, we could provide different probabilities for the infected and recovered populations making our model more accurate.

It's unfortunate but should come as no surprise that there is high variability in the Covid-19 data which should contribute to high variance and as such a higher RMSE or MSE score. This also makes sense because of the data quality associated with Covid-19 tracking. The data tracking for such a wide spread illness is very new and there was a learning curve for much of the data collection for many countries. Most notably was when we tried to train a model that used Spain's Covid-19 data or a lesser developed country's data. For a country like Spain, their recording was inaccurate and wrought with errors. For example, the target variable, new cases for Spain, contained negative new cases values which is impossible. These types of issues with data collection and tracking caused issues when trying to gain insight into a wider variety of countries.

6.2. Final Conclusion

One of the most fascinating parts of training the model and seeing the results was how well it began to perform after handing it fairly few components. The group came in with preconceived notions on what the affecting factors could be, but when trying to regress on four hand picked components, we came up with results that had no meaning. The power of principle component analysis and partial least squares reduction in combining variables to create in relatively few variables was incredible. We managed to learn a lot about regression and dimensionality reduction through this project.

Our results for RMSE for a 30 day prediction were not as good as 4430.89 found by Gupta, et. al using a generalized regression neural network, enhanced by a flower pollinator algorithm (FPA-GRNN) [3], for predicting 50 days in advance. However, we were predicting using a much simpler linear regression model and we trained on only one country's data.

Revisiting our initial motivation, we can see that our current model is simply not accurate enough to compete with state of the art models. While we were unable- at this point in time- to contribute novel research about the importance of vaccines or government policy response, or contribute a new method to model disease, we were able to augment our own learning. We did achieve the goal of better understanding linear regression in a meaningful, applied setting. We ultimately did not learn about which specific input features were most important, but we did experiment with multiple processes for dimensionality reduction and cost functions.

7. Contributions

We worked in pair programming and relied heavily on collaborative brainstorming and coding. We used an agile development process and each week divided up tasks as we defined our discrete objectives. As far as coding contributions, there were many iterations of our models over mulitple .py files. Given that we performed a lot of pair programming and did not equally share driving and navigator roles, the commits on github reflect the work our drivers(active coders) did, but not necessarily the navigator roles.

For the proposal we all contributed to each section, but took the lead on our own section. For the final paper, we roughly distributed the initial draft of the paper into four, and then walked through and edited the entire paper as a group to obtain our final draft. Please find the division of labor for the overall project described below. You can also access our github repos with our individual contributions below:

Sommer:

- 'Introduction', 'Motivation' and 'Contribution' (proposal)
- Figures 1 and 2
- selecting data
- regression attempt 2.py
- LateX formatting and updates (until final iteration) and final paper edits
- Putting together much of the early stages of the presentation
- Individual contributions: Github repo link.

Mitchell:

- 'Evaluation', 'Motivation' and 'Contribution' (proposal)
- 'Plotting Figures 3, 4, 5, 6 '
- Attempt at Scaling Features and Target Values'

- Although I committed and pushed changes to the main branch successfully and they are shown in the commit history, for some reason (possibly pushing from the wrong account) I do not show up as a contributor like Sommer and Ian do. I added a personal github repo for the code that I had worked on in case you have difficulty viewing the commits on the main branch: Github repo link. For these pieces of code, I worked on plotting, scaling and analysis for improvements.
- A lot of results and discussion analysis.

David:

- 'Evaluation', 'Resources', 'Motivation' and 'Contribution' (proposal)
- research formal math equations
- Individual contributions: Github repo link

Ian:

- 'Experiments', 'Resources', 'Motivation' and 'Contribution' (proposal)
- regression attempt 1, pcr.py, gloabl_attempt.py, global_holdout.py
- figure 1,3,4,5,6
- Individual contributions: Github repo link.

The following was our project timeline. This outlines when we accomplished distinct milestones

Week	Date	Goal/Deliverable
Week 1	Oct 22 - Oct 24	Complete Finalized Proposal
	Oct 24 - Oct 24	Submit
Week 2	Oct 25 - Oct 31	Select Dataset and Review Literature
Week 3	Nov 1 - Nov 7	Build baseline
Week 4	Nov 8 - Nov 14	Fine tune and tweak
Week 5	Nov 15 - Nov 21	Fine tune and tweak
Week 6	Nov 22 - Nov 28	Fine tune and tweak
Week 7	Nov 29 - Dec 5	Interpret Results/ Finalize paper

References

[1] S. A. Alanazi, M. Kamruzzaman, M. Alruwaili, N. Alshammari, S. A. Alqahtani, and A. Karime. Measuring and preventing covid-19 using the sir model and machine learning in smart health care. *Journal of healthcare engineering*, 2020, 2020.

[2] J. Farooq and M. A. Bazaz. A deep learning algorithm for modeling and forecasting of covid-19 in five worst affected states of india. *Alexandria Engineering Journal*, 60(1):587–596, 2021.

[3] K. D. Gupta, R. Dwivedi, and D. K. Sharma. Prediction of covid-19 trends in europe using generalized regression neural network optimized by flower pollination algorithm. *Journal of Interdisciplinary Mathematics*, 24(1):33–51, 2021.

[4] L. R.-G. C. A. C. G. E. O.-O. J. H. B. M. D. B. S. D. Hannah Ritchie, Edouard Mathieu and M. Roser. Our world in data. 2021.

[5] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.

[6] L. Harrison. Uk coronavirus (covid-19) machine learning prediction.

[7] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[8] D. P. Kavadi, R. Patan, M. Ramachandran, and A. H. Gandomi. Partial derivative nonlinear global pandemic machine learning prediction of covid 19. *Chaos, Solitons & Fractals*, 139:110056, 2020.

[9] T. pandas development team. pandas-dev/pandas: Pandas, Feb. 2020.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] G. Pinter, I. Felde, A. Mosavi, P. Ghamisi, and R. Gloaguen. Covid-19 pandemic prediction for hungary; a hybrid machine learning approach. *Mathematics*, 8(6):890, 2020.

[12] S. Roberts. All together now: the most trustworthy covid-19 model is an ensemble. *Technology Review*, 2021.

[13] R. Vega, L. Flores, and R. Greiner. Simlr: Machine learning inside the sir model for covid-19 forecasting. *arXiv preprint arXiv:2106.01590*, 2021.