Performance Optimization in Decision Tree Regression Models for Prediction of the COVID-19 Reproduction Rate

Ada Weinert Ravn
Faculty of Mathematics and Natural Sciences, University of Oslo

(Dated: January 3, 2022)

In this project we study the relative prediction accuracy of non-linear regression models on the COVID-19 reproduction rate. Decision tree regression models consider influence of significant features rather than prediction based solely on assumption that past data resembles the future. Three regression models, Random Forest Regressor, Gradient Boosting and XGBOOST that can be used for prediction of the reproduction rate are implemented. 24 features (for example total cases or deaths per million) are ranked using Random Forest Regressor, Gradient Boosting and XGBOOST feature selection algorithms, and nine of them are selected according to the ranks assigned by the above mentioned models. The performance of the respective predictions is evaluated by mean squared error (MSE), Determination Coefficient (R-Squared), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), and the models are tested on the COVID-19 date for Norway and Sweden. Hyperparameter tuning using Grid and RandomSearchCV is applied on the models. We confirm that both the application of feature selection and hyperparameter tuning improves the prediction accuracy. We find the Gradient Boosting regression model to be the best performing with the R-squared score of 0.99369 and MSE of 0.00055 for the Sweden data and the R-squared score of 0.99014 and MSE of 0.0006 for the Norway data being the best achieved results. The most influential features for the application of this algorithm are: total cases, total cases per million, total deaths, total deaths per million and positive rate.

I. INTRODUCTION

An estimation of the rate of spread of the COVID-19 virus is necessary to take relevant preventive measures and precautions allowing us to take control of it. A definite solution as to what factors influence said spread has not yet been found, though it is known that it is highly related to the reproduction rate. "Autoregressive models rely on and work with previous values to forecast future values. Non-linear machine learning regression algorithms have consistently produced the best prediction results in various applications, including the stock exchange, banking, and weather forecasting."[1] By applying such models, the prediction of the factors involved in the spread of the COVID-19 virus becomes possible.

The goal of this project is to explore the performance of the different decision tree regression models when applied in prediction of the reproduction are of the COVID-19 virus. The available features will be evaluated for their application in the Random Forest Regression, Gradient Boosting Regression and XBOOST Regression models, and a set of most important ones will be selected to aid in prediction. The importance and application of hyperparameter tuning for the above models will also be investigated. The models will be implemented and tested on the COVID-19 data for Norway and Sweden, and later assessed using metrics including the Determination Coefficient (R-squared) and Mean Squared Error (MSE). Comparing the application of different models on the same set of data will give us an opportunity to evaluate the relative performance and precision of each of them.

We will first describe the models, methods, and metrics used. The results, along with the discussion of thereof will be presented afterwards, with general comparisons made to each other and other sources. Finally we will finish with a short conclusion based on our findings.

All the code used in this project can be found in the GitHub directory [1].

II. THEORY

A. Reproduction rate

The reproduction number (R), or reproduction rate, indicates the rate of spread of the virus. "The number shows how many people are on average infected by someone who is infected with the corona virus."[2] For R values lower than 1, not enough people are being infected to sustain the outbreak and the spread will eventually stop. However for values above 1 the number of cases will continuously keep increasing. The exact influence of underlying factors on the R number has not yet been defined.

[1] https://github.com/AWRavn/COVID-19-Reproduction-Rate-Prediction

B. Regression using decision trees

This section was heavily cited and paraphrased from the course lecture notes[3].

Decision trees are supervised learning algorithms used for both, classification and regression tasks. the structure of a decision tree resembles a real life tree, being made of nodes and leaves connected by branches. Here a node specifies a test of some attribute of the instance, and the leaf provides the classification of said instance. The branches correspond to possible outcome values of the node tests. The goal of the algorithm is to find the most descriptive features, and then split the data along them such that the feature split is as pure as possible.

The Classification and Regression Tree (CART) algorithm is commonly used for building the tree. It splits the training set in a way that minimizes the mean squared error (MSE).

We have the cost function:

$$C(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right}$$

Here k is a single feature with threshold t_k and m is the node. The MSE for a specific node is defined as:

$$MSE_{node} = \frac{1}{m_{node}} \sum_{i \in node} (\bar{y}_{node} - y_i)^2$$

with

$$\bar{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y_i$$

where $\bar{y}_{node}$ is the mean value of all observations in a specific node.

Single decision trees have high variance which means they often end up overfitting the data. To counteract that ensemble models are used, combining one or more machine learning algorithms. We will discuss and apply three of them: Random Forest Regressor, Gradient Boosting and Extreme Gradient Boosting (XGBoost).

1. Random forest

To improve the variance of the decision tree, we may perform the same operation repeatedly to distinct datasets, a process called bootstrap aggregation. This results in a more accurate, albeit less legible output. In Random Forest Regression model we create a number of decision trees on such bootstrapped training samples. While building the decision trees, at each split we choose a random sample of m predictors from the full set of p predictors - that split is then only allowed to use those m predictors further. A fresh sample of predictors is taken at each split, and typically we choose:

$$m \approx \sqrt{p}$$

This introduces extra randomness when growing decision trees, only searching for the best feature among a small subset of features. This results in a greater tree diversity, and as a consequence, overall more accurate model with lower variance. Here the maximum depth of the tree, as well as the number of trees needs to be tuned to avoid overfitting.

2. Gradient boosting

Gradient Boosting combines so-called weak regressors into a strong method via a series of iterations. We set up weights which will be used to scale to the correctly classified and misclassified cases. The weights are trained sequentially using the residuals from each case - classified to improve prediction. In other words we fit each set of m predictors to the negative gradient values of the cost function.

Given a cost function:

$$C(f) = \sum_{i=0}^{n-1} L(y_i, f(x_i))$$

where i is the index of the observations L is the loss function, y_i is the target and $f(x_i)$ is the function meant to model it. The negative of the cost function is added to our estimate $f_m(x)$, repeated iteratively until we are left with the final estimate $f_M(x)$.

3. XGBOOST

XGBoost or Extreme Gradient Boosting, is an optimized distributed gradient boosting library, providing a more efficient way to use Gradient Boosting for machine learning applications. The details on its function are further described in the "XGBoost: A Scalable Tree Boosting System" article[4].

C. Evaluation Metrics

The performance of each model will be evaluated using metrics describing the difference between the actual

and predicted values. The selected metrics are Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Determination Coefficient (R-Squared Score).

1. Mean Absolute Error

Mean Squared Error (MAE) is a metric calculating the absolute difference between the actual and the predicted values, respectively y_a and y_p. The equation representing it is as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_p - y_a|$$

This metric is robust to outliers, though its output graph is not differentiable. Its data output matches the unit of the output variable.

2. Mean Squared Error

Mean Squared Error (MSE) is similar to MAE, except the square rot of the difference is calculated instead of the absolute difference. The equation is as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_p - y_a)^2$$

The advantage of MSE over MAE is its differentiable graph, allowing easy application a loss function. As a downside it heavily penalizes outliers.

3. Root Mean Squared Error

Root Mean Squared Error (RMSE) is the root of MSE. It is represented by the following equation:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_p - y_a)^2}$$

It is a commonly used metric with the output value with the same unit as the data output making interpretation easy. It it less robust to outliers as compared to MAE and is used when the error is expected to be highly non-linear.

4. Determination Coefficient

As opposed to the other metrics described above, the Determination Coefficient, also called R-Squared Score, is not a loss, but rather a metric representing the performance of the model. The equation for it is as follows:

$$R^2 = 1 - \frac{RSS}{TSS}$$

Here RSS is the sum of squares of residuals, or error of the regression line, and TSS is the total sum of squares, or error of mean line. An R score of 1 would indicate the perfect prediction.

III. COVID-19 DATASET

The recent[2] global Covid-19 data as maintained by Our World In Data[3] [5] was used for the study of the estimated reproduction rate. From 207 available country profiles Norway and Sweden have been selected, with records spanning between 22/02/2020 and 27/12/21, and sorted by date. Additional preprocessing was applied to account for the empty cells, and entries missing associated with reproduction rate have been omitted. A total of 24 features have been used in initial feature selection process: total cases, new cases, total deaths, new deaths, total cases per million, new cases per million, total deaths per million, new deaths per million, new tests, total tests, total tests per thousand, new tests per thousand, positive rate, tests per case, stringency index, population density, total vaccinations, people vaccinated, people fully vaccinated, total boosters, total vaccinations per hundred, people vaccinated per hundred, people fully vaccinated per hundred and total boosters per hundred. This resulted in 661 total data points, each represented by 24 features for each of the two countries.

IV. IMPLEMENTATION

The models were implemented and evaluated using Python's Sci-Kit Library [4], with the exception of XGBOOST[5] which was imported from its own namesake library. The initial implementation of the models is run

[2] Available as of 31/12/22.

[3] Available online at: https://github.com/owid/covid-19-data/tree/master/public/data

[4] Link to documentation: https://scikit-learn.org/stable/modules/classes.html

[5] Link to documentation: https://xgboost.readthedocs.io/en/stable/

with the default parameters held by each model, and all seeds in the program are set to 43.

The hyperparameters for the Gradient Boosting Regression and Random Forest Regression were tuned using GridSearchCV model with the following parameter values: scoring='neg_mean_squared_error', cv=3 and n_jobs=-1. For the XGBOOST Regression RandomizedSearchCV was used instead, with parameter values: scoring='neg_mean_squared_error', cv=3, n_jobs=-1 and n_iter=100. The exact values included in the search can be seen in the GitHub directory. Due to the time constraints only the Gradient Boosting Regression and Random Forest Regression were extensively tuned, but a simple pass on tuning XGBOOST has also been made, and can be further improved using the implementation.

The importance scores have been first evaluated by the importance of their relative weight, contained within their respective model implementations. Afterwards a permutation based importance scores have also been found using the Sci-Kit Library. A limited selection of most relevant features have then been found based on results, general evaluation and partially trial-and-error.

The dataset was split so that 80% have been used for training and the remaining 20% for testing and the data was scaled before the analysis using StandardScaler() from the Sci-Kit Library. Some additional support for the dataset processing has been partially implemented but is otherwise beyond the scope of this project.

V. RESULTS AND DISCUSSION

A. Feature selection

The feature importance scores obtained by Random Forest Regression, Gradient Boosting Regression and XGBOOST Regression are given in detail in Section VI: in Table VI and VII and plotted in Figure 5, 6, 7 and 8. A quick overview over the relative feature importance can also be seen in Figure 1 and 2.

From the data we can see that the feature importance varies between the countries and between the different methods. Overall the algorithms for the Random Forest Regression and Gradient Boosting Regression seem to prioritize similar features, while XGBOOST Regression strongly focuses on a few characteristics and to a large degree disregards the others. Features such as total cases and total cases per million are consistently prioritised, while new tests and total deaths are prioritized higher in Norway and Sweden respectively. This indicates differences in the significance of feature influence on the reproduction rate based on the country.

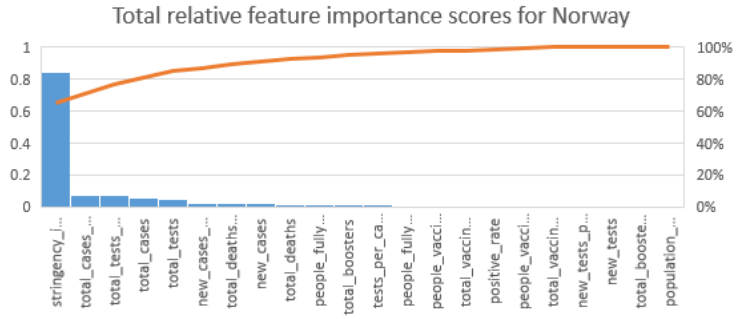

Figure 1. Representation of relative feature importances across all methods for Norway.

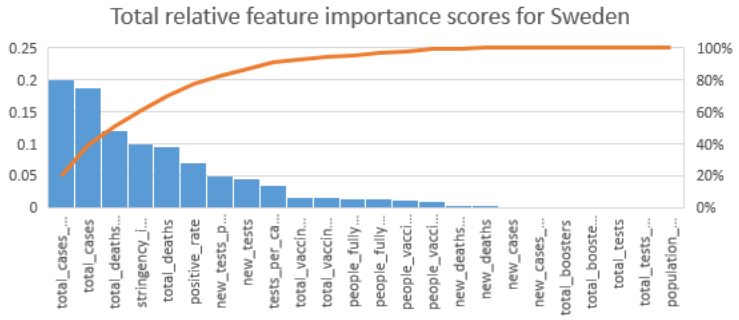

Figure 2. Representation of relative feature importances across all methods for Sweden.

The stringency index in particular is a consistently prioritised feature. It is a numeral estimate of of the measures taken to combat the spread of the virus, such as school closures and travel bans. The factor is particularly highly weighted for Norway, where close downs and restriction closely followed changes in the number of cases. While the correlation clearly makes sense it has been excluded from the consideration due to being based on many of the same initial factors as the reproduction rate itself.

Increasing the number of features might generally improve the performance of the model at the cost of computation speed. However it also decreases the diversity of individual trees which is an important selling point of the algorithm. Careful selection and testing is necessary to achieve optimal results.

Based on the discussion, out of the 24 features, 9 were identified as most influential with regards to the reproduction rate - a common set for both countries and all methods: total cases, total deaths, total cases per million, total deaths per million, new tests, total tests, tests per case, positive rate, and people fully vaccinated.

	Metric	Method		
		Random Forest Regression	Gradient Boosting Regression	XGBoost Regression
Norway	MSE	0.00185	0.00467	**0.00169**
	MAE	0.03155	0.05307	**0.03104**
	RMSE	0.04296	0.06831	**0.04108**
	R-squared	0.96962	0.92320	**0.97223**
Sweden	MSE	**0.00081**	0.00359	0.00146
	MAE	**0.01869**	0.04364	0.02622
	RMSE	**0.02847**	0.05990	0.03822
	R-squared	**0.99068**	0.95875	0.98321

Table I. Metrics for prediction without feature selection and hyperparameter tuning.

B. Hyperparameter tuning

Before presenting the optimised parameters let us first briefly explain how each of them is expected to affect the output. We will here limit the discussion to parameters influencing the output, for discussion about general implementation see Section IMPLEMENTATION.

Three parameters were the goal of optimization of Random Forest Regression in this project: n_estimators, max_depth and min_samples_split. Number of decision trees, here called n_estimators, describes the amount of trees we want to build before taking the maximum voting or averages of predictions. While the higher the number of trees makes the predictions more stable and accurate, it comes at a cost of computation speed, and is expected to quickly hit the point of diminishing returns. Each node is expanded to the max_depth depth of the tree and min_samples_split is the minimum number of samples required to split a node. Hence the parameters are related to each other.

	Metric	Method		
		Random Forest Regression	Gradient Boosting Regression	XGBoost Regression
Norway	MSE	0.00125	**0.00086**	0.00108
	MAE	0.02399	**0.02159**	0.02478
	RMSE	0.03537	**0.02938**	0.03287
	R-squared	0.97940	**0.98579**	0.98222
Sweden	MSE	**0.00084**	0.00316	0.00087
	MAE	**0.01776**	0.04102	0.02047
	RMSE	**0.02891**	0.05623	0.02951
	R-squared	**0.99039**	0.96365	0.98999

Table II. Metrics for prediction with feature selection and without hyperparameter tuning.

Following that, four parameters were the goal of optimization of Gradient Boosting Regression in this project: n_estimators, max_depth, subsample and learning_rate. Just like above n_estimators is the number of trees, here it is more robust to overfitting and using a larger number should result in a better performance. learning_rate shrinks the contribution of each tree, with lower values making the model more robust, allowing it to generalize well. However lower values require a higher number of trees and are more computationally expensive. n_estimators tends to be tuned to a particular learning_rate as a result. subsample indicates the fraction of observations to be selected for each tree, with value lower than 1 leading to reduction in variance and increase in bias. Finally max_depth defines the depth of the tree and is used to control overfitting, with the best value depending on the input variables.

Finally the XBOOST regression uses the same parameters as Gradient Boosting Regression with addition of cosample_bytree which is the subsample ratio of columns when constructing each tree. A notable difference is that max_depth, which increases the complexity of the model, is likely to overfit for values above 6. Only one instance of this optimization was run due to time constraints.

The best tuned hyperparameters found within available timeframe for each method are listed in Table III. There is potential to improve the scores using either higher amount of iterations and parameters or by using extensive grid search algorithm in case of XGBOOST Regression.

		Norway		Sweden	
Method	Parameter	max features	min features	max features	min features
Random Forest Regression	max_depth	100	150	150	100
	min_samples_split	2	2	2	2
	n_estimators	500	500	2000	2000
Gradient Boosting Regression	learnining_rate	0.01	0.01	0.01	0.01
	max_depth	6	12	10	12
	n_estimators	2000	2000	2000	2000
	subsample	1	0.75	0.5	0.75
XGBoost Regression	learnining_rate	0.1	0.1	0.1	0.1
	max_depth	5	5	5	5
	n_estimators	1000	1000	1000	1000
	subsample	0.5	0.5	0.5	0.5
	cosample_bytree	0.3	0.3	0.3	0.3

Table III. Best tuned values of the hyperparameters for the different regression methods. Max features and min features refer to the whole feature set and the selected feature set respectively.

While most of the hyperparamters remain constant across the applications for each of the four sets of data, relatively high variance in max_depth and subsample can be observed. More experiments will be necessary to correlate those changes with specific case scenarios. Overall all values appear to be within expected constraints.

	Metric	Method		
		Random Forest Regression	Gradient Boosting Regression	XGBoost Regression
Norway	MSE	0.00198	**0.00163**	0.00193
	MAE	0.03259	**0.02924**	0.03184
	RMSE	0.04451	**0.04042**	0.04396
	R-squared	0.96739	**0.97312**	0.96819
Sweden	MSE	0.00087	**0.00055**	0.00089
	MAE	0.01896	**0.01592**	0.01919
	RMSE	0.02948	**0.02343**	0.02978
	R-squared	0.99001	**0.99369**	0.98980

Table IV. Metrics for prediction without feature selection and with hyperparameter tuning.

C. Prediction performance

We perform the experiments using Random Forest, Gradient Boosting and XGBOOST Regression model applied for reproduction rate prediction. We want to achieve the most accurate prediction while investigating the impacts of feature selection and hyperparameter tuning. A total of four cases were implemented with and without either parameter tuning or feature selection. The results were then evaluated using mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE) and determination coefficient (R-squared). We found that the evaluation metrics correlated with one another and will as such be discussed together. The best and worst result of each case as well as of the project as a whole will be discussed below. The results will be compared to those found in the relevant research paper published in the Frontiers in Public Health[1] about five months prior. For reference that study applied 16 initial and 7 selected features respectively, but does not specify in detail how the input data was handled, and which country or countries the estimation was based on.

The first case tests the performance of the model using all default hyperparameters for each model, and using all 24 available features. The resulting performance metrics can be found in Table I. Here the results are different for each country. For Sweden data the Random Forest Regression is the most successful with the R-squared score of 0.99068 and MSE of 0.00081, which is also the third overall best score found. For Norway the best performing algorithm was XGBOOST Regression, which makes sense since that's the dataset it was modelled after, with the R-squared score of 0.97223 and MSE of 0.00169. The worst performing metric here was Gradient Boosting Regression, in particular for the Norway dataset, with R-squared of 0.92320 and MSE of 0.00467. This was also the overall worst score for this project. The best result of R-squared equal to 0.99068 is higher than compared best result of 0.97923 in the reference paper, both using the Random Forest Regression algorithm. This implied that said model is great choice if you can't optimize any of the inputs.

The second case tests the performance of the model using all default hyperparameters for each model, and applying the feature selection. The resulting performance metrics can be found in Table II. Here the best case is different for each country as well, with Sweden once again leading with the application or Random Forest Regression achieving the R-squared score of 0.99039 and MSE of 0.00084. For Norway the best score was achieved using the Gradient Boosting Regression with R-square score of 0.98579 and MSE of 0.00086. This is also the second best score for Norway dataset overall. The worst score was achieved for the Gradient Boosting Regression for Sweden with R-squared score of 0.96365 and MSE of 0.00316. This indicates that the performance of the Gradient Boosting Regression model is highly variable with regards to data and/or feature selection. Compared to the prior case we can see good to great improvement of performance with addition of feature selection with exception of Random Forest Regression model for Sweden. Once again we beat the best score of R-squared of 0.97988 in the reference paper, achieved using Gradient Boosting Regression model.

	Metric	Method		
		Random Forest Regression	Gradient Boosting Regression	XGBoost Regression
Norway	MSE	0.00125	**0.00060**	0.00108
	MAE	0.02399	**0.01714**	0.02478
	RMSE	0.03537	**0.02448**	0.03287
	R-squared	0.97940	**0.99014**	0.98222
Sweden	MSE	0.00088	**0.00059**	0.00069
	MAE	0.01807	**0.01534**	0.01759
	RMSE	0.02974	**0.02431**	0.02620
	R-squared	0.98983	**0.99321**	0.99211

Table V. Metrics for prediction with feature selection and hyperparameter tuning.

The third case tests the performance of the model using tuned hyperparameters for each model, and using all 24 available features The resulting performance metrics can be found in Table IV. Here the Gradient Boosting Regression model is overall most successful for both datasets, resulting in R-squared score of 0.97312 and MSE of 0.00163 for Norway and R-squared score of 0.99369 and MSE of 0.00055 for Sweden. The later being the overall best score for this project. The worst performing model here was the Random Forest Regression on Norway dataset with R-squared score of 0.96739 and MSE of 0.00198. Clear improvement can be observed over the first case test and

the improvement is comparable to that achieved by feature selection. In the reference paper the best scoring algorithm was Random Forest Regression with R-squared score of 0.97637, which was worse than the achieved result.

Finally in the last case we test the performance of the model using the tuned hyperparameters for each model, and applying the feature selection. The resulting performance metrics can be found in Table II. Once again the Gradient Boosting is the best performing model for both datasets, with R-squared score of 0.99321 and MSE of 0.00059 for Sweden being the second best score overall, and R-squared score of 0.99014 and MSE of 0.00060 for Norway being the overall best score for Norway. Across the methods and datasets this is the highest scoring input combination, although not all individual scores are superior. The scores achieved are superior to those obrained in the reference paper, namely R-squared score of 0.97637 for the Random Forest Regression. That is our worst performing algorithm here is with R-squared score of 0.97940 and MSE of 0.00125.

We observe that Random Forest Regression appears to be a very low-floor, high-ceiling algorithm which often either performs best or worst. The Gradient Boosting Regression works best with inclusion of hyperparameters and is the overall most successful model. XGBOOSTING Regression, while included, did not have properly optimised parameters for many cases, but it performed great where it was properly optimized. It can be argued that it can potentially achieve better scores that the current best. Overall better scores are observed for the Sweden dataset, but the inclusion of hyperparameter tuning and feature selection makes overall most difference for the Norway dataset, which is only worse by approximately 0.003 in temrs of the R-squared score.

The difference in result as compared to the reference paper can be attributed to additional 6 months of relevant data, wider range of available features or simply target country. Nonetheless we find that both hyperparameter tuning and careful feature selection improve the prediction of the reproduction rate. Additionally as already implied[1], the selected features suggest that total deaths and total deaths per million strongly affect the prediction rate instead of only depending on past predictor variable.

The predicted and actual values are plotted in Figure 3 and 4 for best solution for each of the datasets. We can see that the predicted values lay very close to the actual values.

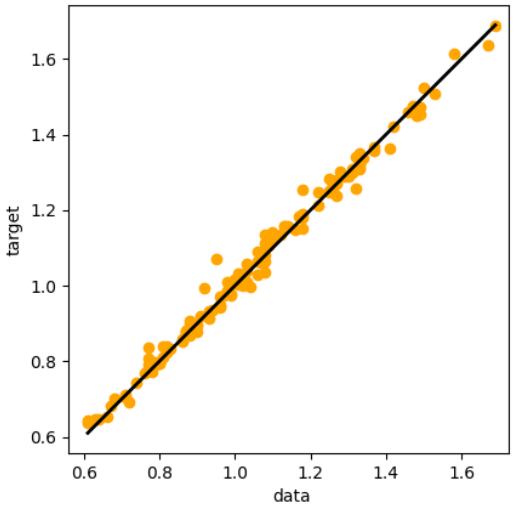

Figure 3. Representation of the predicted value against the actual value for best case for Norway. Achieved using the Gradient Boosting Regression with feature selection and hyperparameter tuning.

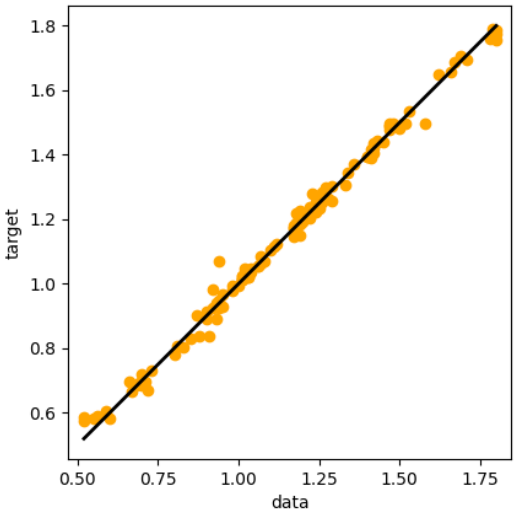

Figure 4. Representation of the predicted value against the actual value for best case for Sweden. Achieved using the Gradient Boosting Regression with hyperparameter tuning and no feature selection.

VI. CONCLUSION

In this project we have studied the relative prediction accuracy of non-linear regression models on the COVID-19 reproduction rate and parameters affecting it.

We have discovered that when applied on the local data, both hyperparameter tuning and feature selection result in a significant improvement in prediction accuracy. Out of the 24 features considered, 9 were proven to be prominent in reproduction rate prediction, namely:

total cases, total deaths, total cases per million, total deaths per million, new tests, total tests, tests per case, positive rate, and people fully vaccinated. The Gradient Boosting Regression model, which is the overall most accurate in the scope of this project, placed highest importance on total cases, total cases per million, total deaths, total deaths per million and positive rate as seen on Figure 9. We have also discovered that while the Random Forest Regression model can yield high scores it is just as likely to provide low accuracy scores, so using it alongside other models is advised.

Further work should include a more in-depth investigating of the XGBOOSTING hyperparaneter selection, as well as attempt to apply the models on the global population. There is potential to include an even wider array of potential features when investigating less homogenous populace, and their importance in that case needs to be evaluated.

REFERENCES

[1] J. Kaliappan, K. Srinivasan, S. Mian Qaisar, K. Sundararajan, C.-Y. Chang, and S. C, "Performance evaluation of regression models for the prediction of the covid-19 reproduction rate," *Frontiers in Public Health*, vol. 9, p. 1319, 2021. https://www.frontiersin.org/article/10.3389/fpubh.2021.729795.

[2] https://coronadashboard.government.nl/landelijk/reproductiegetal.

[3] M. Hjorth-Jensen, "Applied data analysis and machine learning, fys-stk3155/4155 at the university of oslo, norway," 2021.

[4] T. Chen and C. Guestrin, "Xgboost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016.

[5] L. R.-G. C. A. C. G. E. O.-O. J. H. B. M. D. B. Hannah Ritchie, Edouard Mathieu and M. Roser, "Coronavirus pandemic (covid-19)," *Our World in Data*, 2020. https://ourworldindata.org/coronavirus.

APPENDIX

Feature	Norway			Sweden		
	Random Forest Regression	Gradient Boosting Regression	XGBoost Regression	Random Forest Regression	Gradient Boosting Regression	XGBoost Regression
total_cases	0.17265	0.18465	0.05368	0.18724	0.18332	0.13005
new_cases	0.02745	0.01664	0.00863	0.00249	0.00085	0.00116
total_deaths	0.10935	0.08112	0.00032	0.09572	0.13742	0.52246
new_deaths	0.00157	0.00363	0.00282	0.00347	0.00108	0.00441
total_cases_per_million	0.21637	0.18887	0.00000	0.19978	0.24367	0.00000
new_cases_per_million	0.03107	0.03535	0.00000	0.00193	0.00030	0.00000
total_deaths_per_million	0.10540	0.13371	0.00000	0.12062	0.10952	0.00000
new_deaths_per_million	0.00151	0.00220	0.00000	0.00395	0.00048	0.00000
new_tests	0.01251	0.01053	0.00213	0.04603	0.02547	0.05580
total_tests	0.03395	0.03076	0.78355	0.00000	0.00000	0.00000
total_tests_per_thousand	0.04177	0.04624	0.00000	0.00000	0.00000	0.00000
new_tests_per_thousand	0.01313	0.01002	0.00000	0.04956	0.02874	0.00000
positive_rate	0.01441	0.00866	0.00209	0.07135	0.06182	0.03698
tests_per_case	0.01663	0.03496	0.00500	0.03517	0.02389	0.00495
total_vaccinations	0.00869	0.01334	0.00000	0.01600	0.00261	0.05359
people_vaccinated	0.01333	0.00700	0.00000	0.01139	0.01150	0.00000
people_fully_vaccinated	0.01313	0.00963	0.00000	0.01432	0.00577	0.00000
total_boosters	0.01085	0.01549	0.00000	0.00006	0.00000	0.00000
total_vaccinations_per_hundred	0.00871	0.01229	0.00000	0.01714	0.00350	0.00000
people_vaccinated_per_hundred	0.00954	0.00995	0.00000	0.01015	0.00656	0.00000
people_fully_vaccinated_per_hundred	0.01119	0.00497	0.00000	0.01453	0.01967	0.00000
total_boosters_per_hundred	0.00236	0.00104	0.00000	0.00005	0.00000	0.00000
stringency_index	0.12443	0.13895	0.14179	0.09905	0.13384	0.19060
population_density	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table VI. Feature importance scores according to their relative weights for each algorithm.

Feature	Norway			Sweden		
	Random Forest Regression	Gradient Boosting Regression	XGBoost Regression	Random Forest Regression	Gradient Boosting Regression	XGBoost Regression
total_cases	0.05731	0.07504	0.87243	0.12198	0.13039	1.13349
new_cases	0.02301	0.01122	0.12056	0.00020	0.00081	-0.00032
total_deaths	0.01850	0.03119	-0.00003	0.05801	0.08395	0.00244
new_deaths	-0.00002	-0.00314	0.00650	0.00014	0.00040	0.01499
total_cases_per_million	0.07616	0.12428	0.00000	0.12137	0.12034	0.00000
new_cases_per_million	0.02722	0.07081	0.00000	0.00022	0.00027	0.00000
total_deaths_per_million	0.02430	0.02943	0.00000	0.07519	0.13306	0.00000
new_deaths_per_million	-0.00036	-0.00267	0.00000	0.00043	0.00045	0.00000
new_tests	0.00232	0.00305	0.00468	0.04455	0.02320	0.10920
total_tests	0.04963	0.07707	0.02939	0.00000	0.00000	0.00000
total_tests_per_thousand	0.07505	0.07463	0.00000	0.00000	0.00000	0.00000
new_tests_per_thousand	0.00239	0.00168	0.00000	0.04943	0.04181	0.00000
positive_rate	0.00858	0.00477	0.01986	0.11988	0.07004	0.17339
tests_per_case	0.01273	0.11135	0.02157	0.01318	0.01484	0.00447
total_vaccinations	0.00675	0.01673	0.00000	0.00828	0.00140	0.02045
people_vaccinated	0.00941	0.00467	0.00000	0.00425	0.00843	0.00000
people_fully_vaccinated	0.01598	0.02021	0.00000	0.00765	0.00757	0.00000
total_boosters	0.01589	0.02759	0.00000	0.00000	0.00000	0.00000
total_vaccinations_per_hundred	0.00889	0.04169	0.00000	0.00810	0.00351	0.00000
people_vaccinated_per_hundred	0.00815	0.01190	0.00000	0.00371	0.00641	0.00000
people_fully_vaccinated_per_hundred	0.01028	0.01310	0.00000	0.00761	0.01052	0.00000
total_boosters_per_hundred	0.00123	0.00167	0.00000	0.00000	0.00000	0.00000
stringency_index	0.83918	0.56913	0.67714	0.20459	0.39093	0.13286
population_density	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table VII. Feature permutation importance scores according to their relative weights for each algorithm.

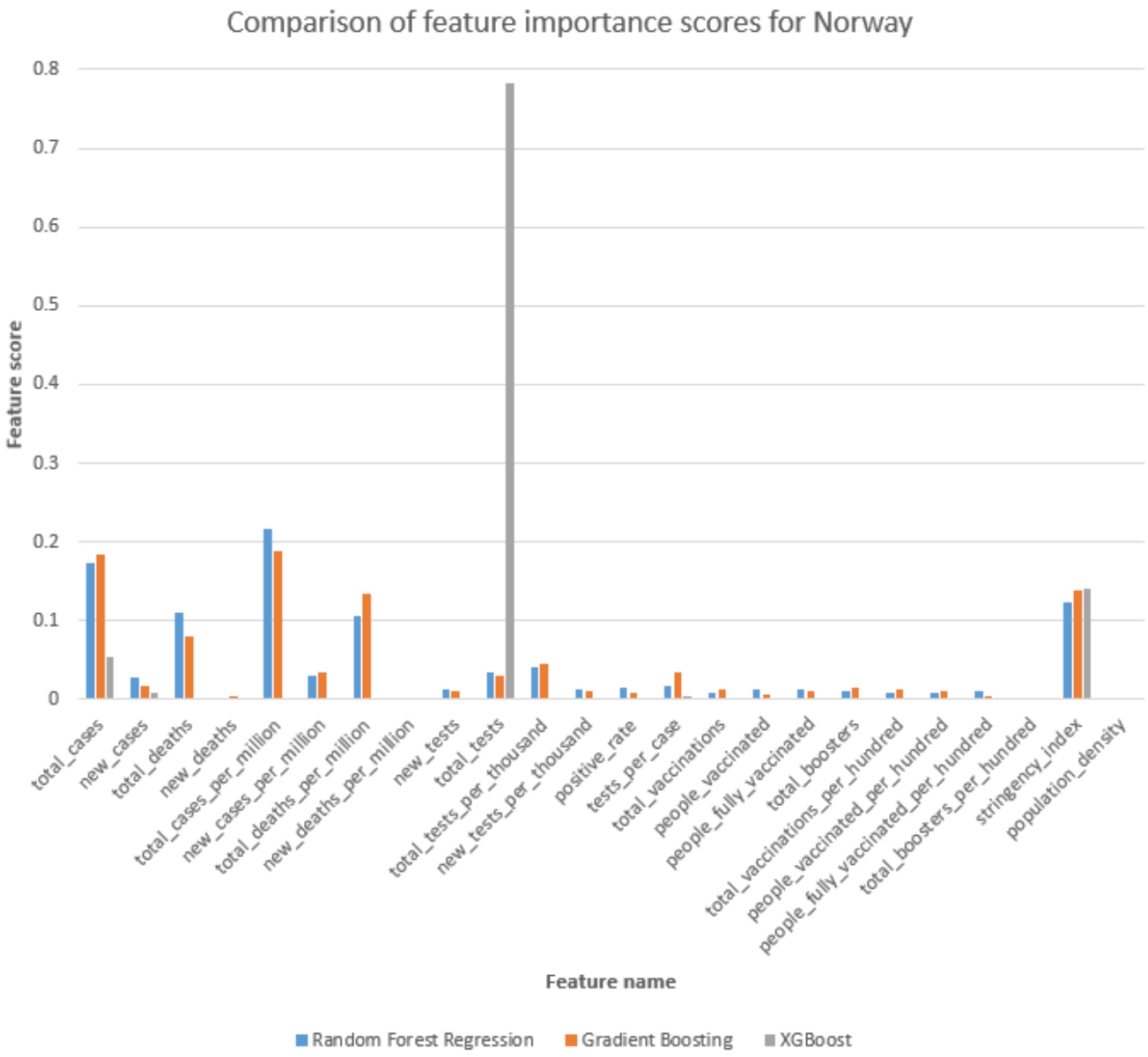

Figure 5. Comparison graph of feature importance scores for each algorithm.

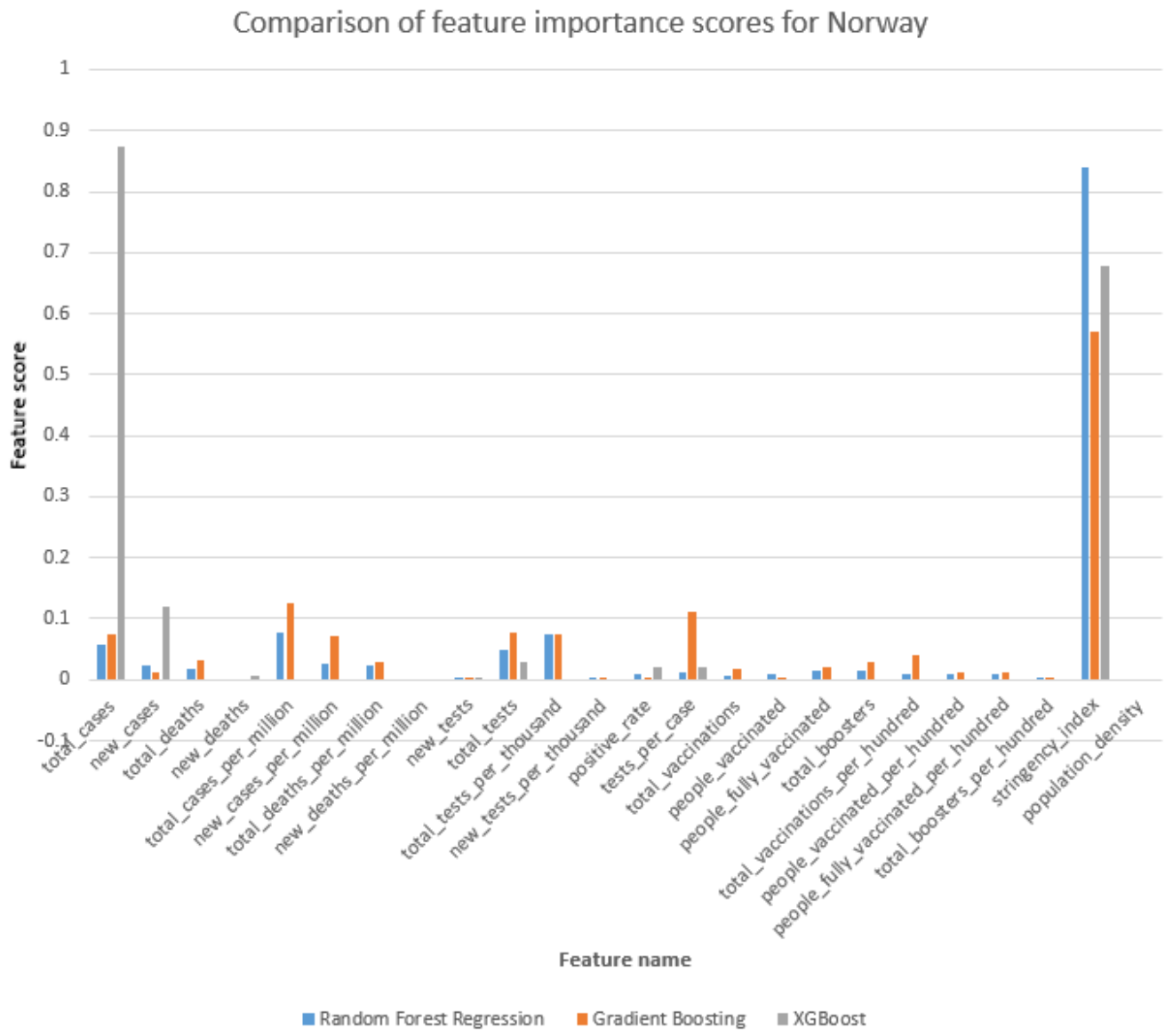

Figure 6. Comparison graph of feature permutation importance scores for each algorithm.

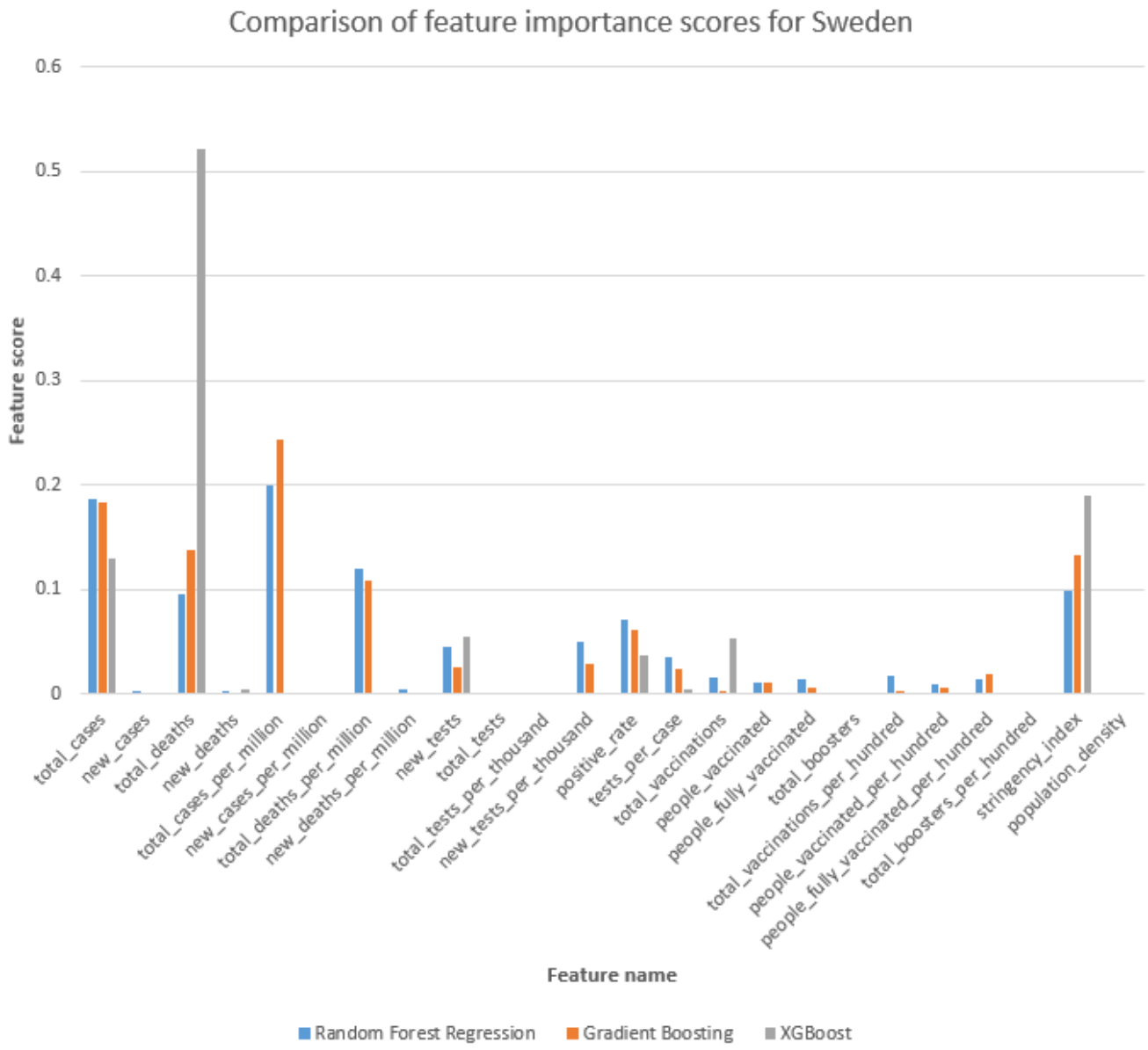

Figure 7. Comparison graph of feature importance scores for each algorithm.

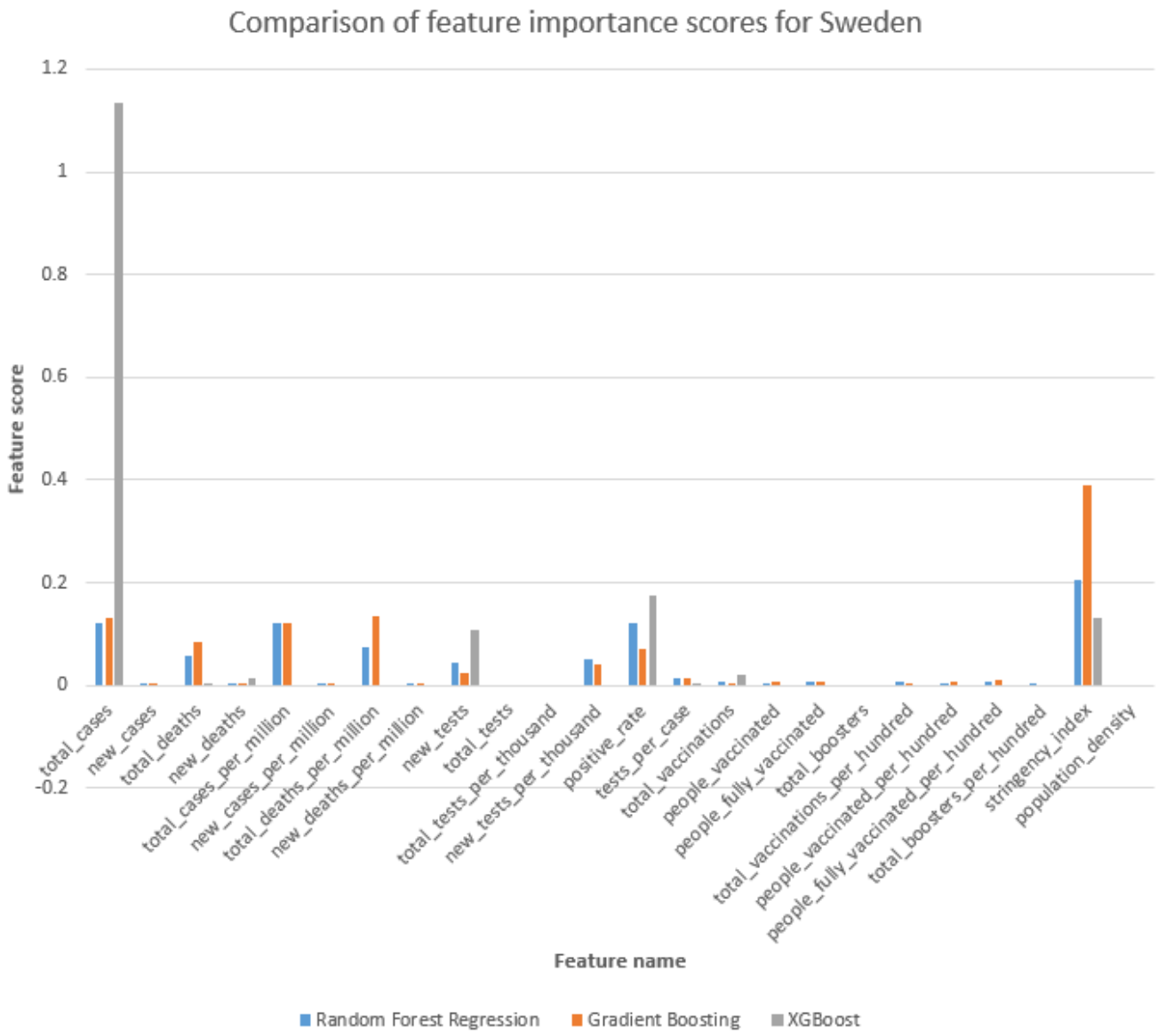

Figure 8. Comparison graph of feature permutation importance scores for each algorithm.

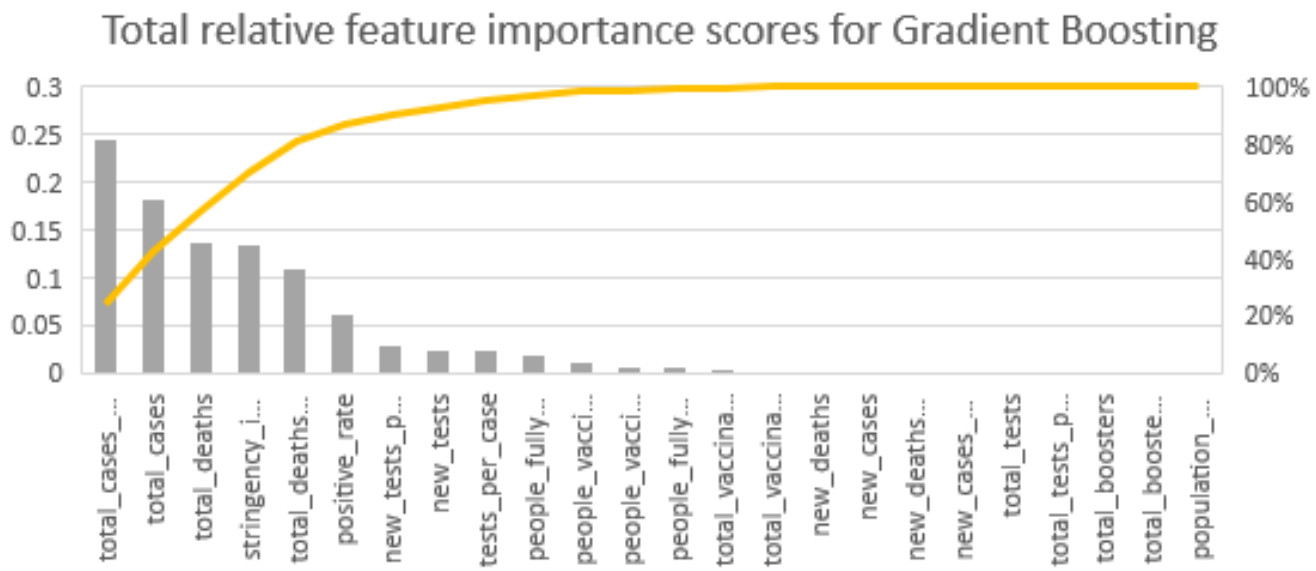

Figure 9. Representation of relative feature importance for the Gradient Boosting Regression model.