Term Project Phase 3

Aya Mourad

December 10, 2020

Hypothesis Testing: Comparing Regression Models

In phase 3 of the project, we evaluated our models using k-fold cross-validation. However, this approach can be misleading as it could be hard to identify whether the mean of the MSE resulted from the cross validation is real or is a statistical fluke. To address this problem, statistical significance tests can be used to improve the confidence in the explanation and the presentation of results when selecting the best model.

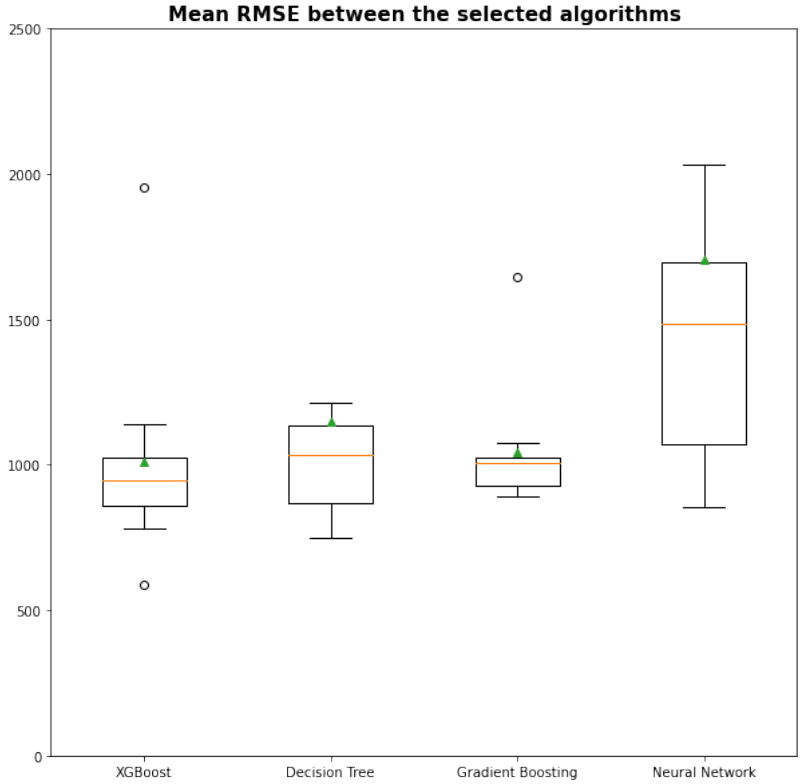
We have evaluated our models on 10 k-fold cross validation and calculated the MSE for each split. Each model gave us a results list containing 10 MSEs for 10-fold cross validation. One hypothesis testing that can be done is Student's t-test to check if the mean MSEs between the models is statistically significant. However, the observations in each sample is not independent, the k-fold cross validation procedure requires using same observation in the training dataset K-1 times i.e. same rows of data are trained multiple times. Thus, the 10fold MSEs are dependent.

To evaluate our models,we used two-fold cross-validation with five repeats (5x2- fold cross validation) and paired students t-test (dependent data). Repeated meaures Anova (for dependent data) can be also suitable in evaluating our models, however we didn't find its library and it is not yet available in Scipy.

Paired ttest 5x2cv

We used MLxtend Library in python using paired_ttest_5x2cv() function. This function reports the t-statistic value and the p-value to indicate whether the difference of 2 models performance is statistically significant or not. We have interpreted the p-value with alpha equals to 5% (0.05). If the p-value is less or equal to alpha, we reject the null hypothesis that the models have the same mean performance which means that the difference is probably real. However, if the p-value is greater than alpha, we

fail to reject the null hypothesis meaning the models have same mean performance and the observed difference in the mean MSEs is probably a statistical fluke.



According to the mean RMSE resulted from 10-fold cross validation, XGBoost is the best model whereas Neural Network is the worst.
The results of the paired_ttest_5x2cv() gives us a p-value to indicate whether the mean MSEs of the 10fold cross validation is significant or not, the p-value reported when comparing XGBoost and Decision tree is 0.72 and the p-value when comparing XGBoost and Gradient Boosting model is 0.994 . Both values are much larger than 0.05 which leads us to fail to reject the null hypothesis, suggesting that any observed difference between the model is probably not real which highlights that performing model selection based only on the mean performance may not be sufficient. The three models perform the same on average, we chose XGBoost as our best model among the three models and compared it to the Neural Network Model.

We compared XGBoost with the neural network algorithm the p-value is 0.038 which is less than 0.05 this leads us to to reject the null hypothesis, suggesting that any observed difference between the algorithms is probably real. Thus, XGBoost performs better than the Neural Network model.

Paired student's test

Because the 10-fold MSEs are not independent, we cannot use the Student's t-test. Instead, we must use paired Student's t-test that handles dependent data samples. The default assumption, or null hypothesis of the test, is that there is no difference in the means between the samples. The rejection of the null hypothesis indicates that there is enough evidence that the sample means are different.

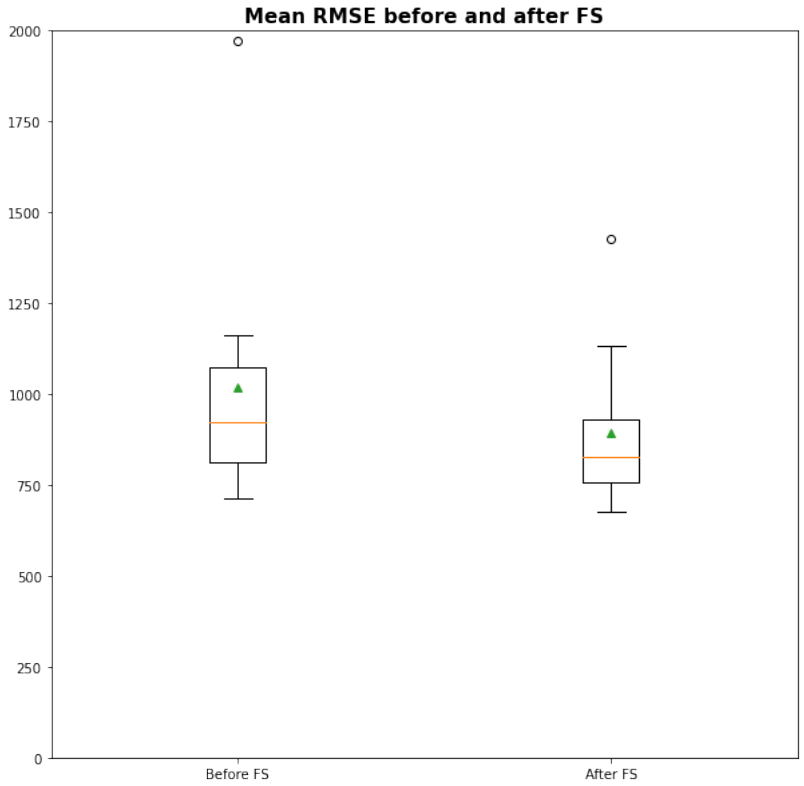
- Fail to Reject H0: Paired sample distributions are equal.
- Reject H0: Paired sample distributions are not equal.

Comparing XGBoost with decision tree and gradient boosting gives us a p-value equal 0.494 and 0.178 respectively which is higher than 0.05, this leads us to fail to reject H0 suggesting that the models have no difference in the means of their MSEs. However, when comparing XGBoost with Neural network the p-value was 0.004 which leads us to reject H0 meaning that we have enough evidence that the means of MSEs are different i.e. not statistical fluke and XGBoost performance is better than the Neural Network Model.

Thus, XGBoost, decision tree, or gradient boosting can be chosen as our best model. We have chosen XGBoost and perform pre-post feature selection and SHAP using it.

Hypothesis Testing: Best Model with Feature Selection

Since the results of feature selection are done on different columns of the dataset, we can't use (5x2 fold cross validation) as the function requires to pass the X to evaluate two models, and in our case, the X differs pre and post feature selection making it impossible to compare the models using this test. Although the X_train dataset is different pre and post feature selection, however, both datasets are some how related. The main difference between both datasets is the later (post fs) have fewer number of columns, whereas we are still evaluating the same sample row size using the same model which suggests that our data samples are dependent. To perform hypothesis testing, we used paired student's test.



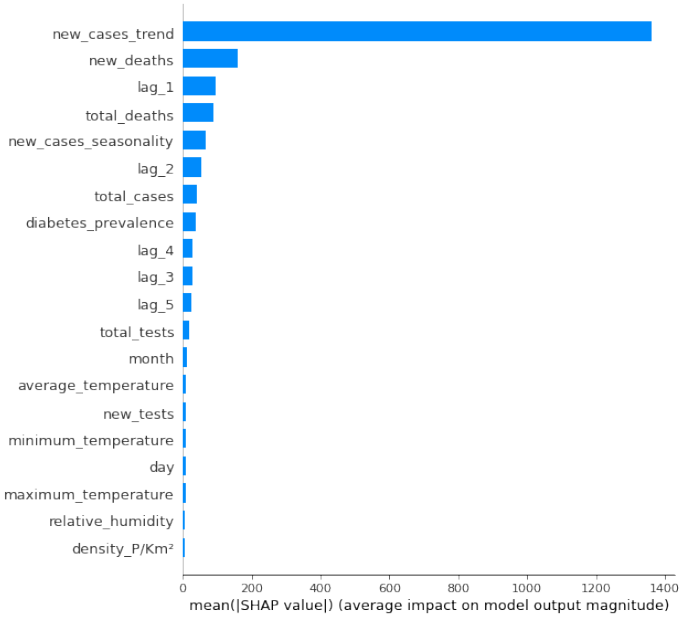
The mean MSE improved post feature selection. We evaluated statistically this evaluation using paired student's test, the resulted p-value was 0.124 which is greater than alpha (0.05). this leads us to fail to reject H0 suggesting that the models have no difference in the means of their MSEs. Thus, feature selection statistically has no significance impact on the XGBoost model.

SHAP Interpertibality

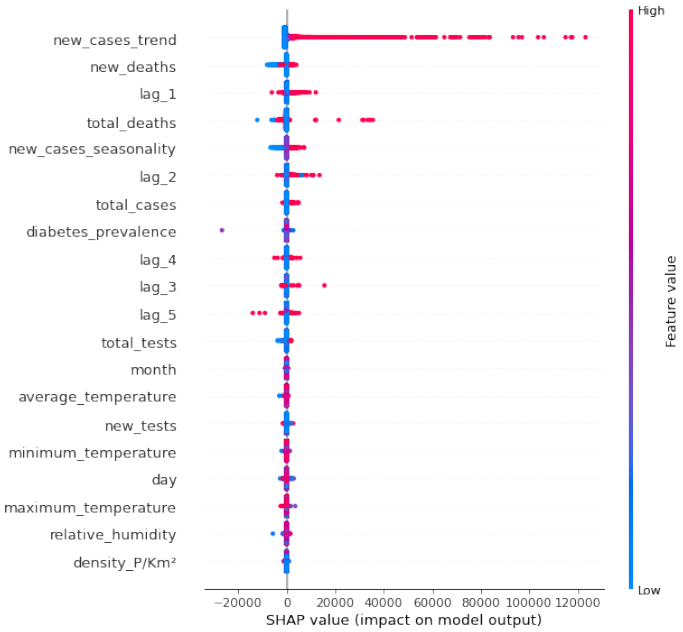
Global Interpretability Plots

Variable Importance Plot

A variable importance plot shows the most important variables in descending order. The top features have an impact on the model more than the bottom ones and therefore have a higher predictive power. Below we can see the top features of our model including new_cases_trend, lag_2, new_deaths, lag_5, total_deaths and new_cases_seasonality as top features of our model.



Below is SHAP value plot that shows the positive and negative relationships of the features with our target variable (new_cases)

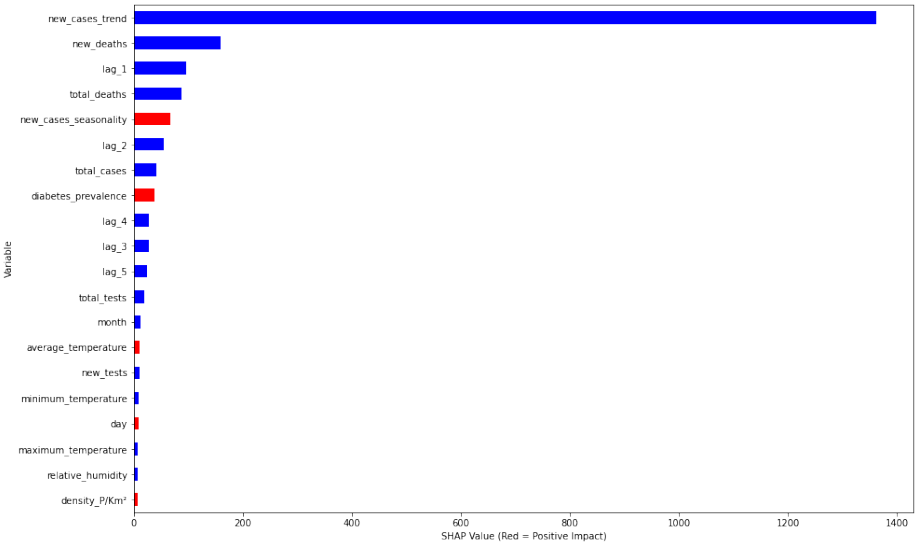


Variable Importance Plot

The plot shows the following:

- Features are ranked in descending order according to their importance
- Impact (x-axis) shows the effect of the features value on the expected value of new cases (positive or negative numbers). Whereas, blue color indicates low correlation and red shows high correlation. For example, A high level of the 'new_cases_trend' has a positive effect on new_cases (positive impact on x-axis) and highly correlated with new cases (red color). Similarly, we can say the new_case_seasonality is negatively correlated with the target variable (new_cases)

To simplify the above plot, we have plotted the top features in the descending order and computed the correlation between the shap_values and the features where red color shows high correlation with the target variable.



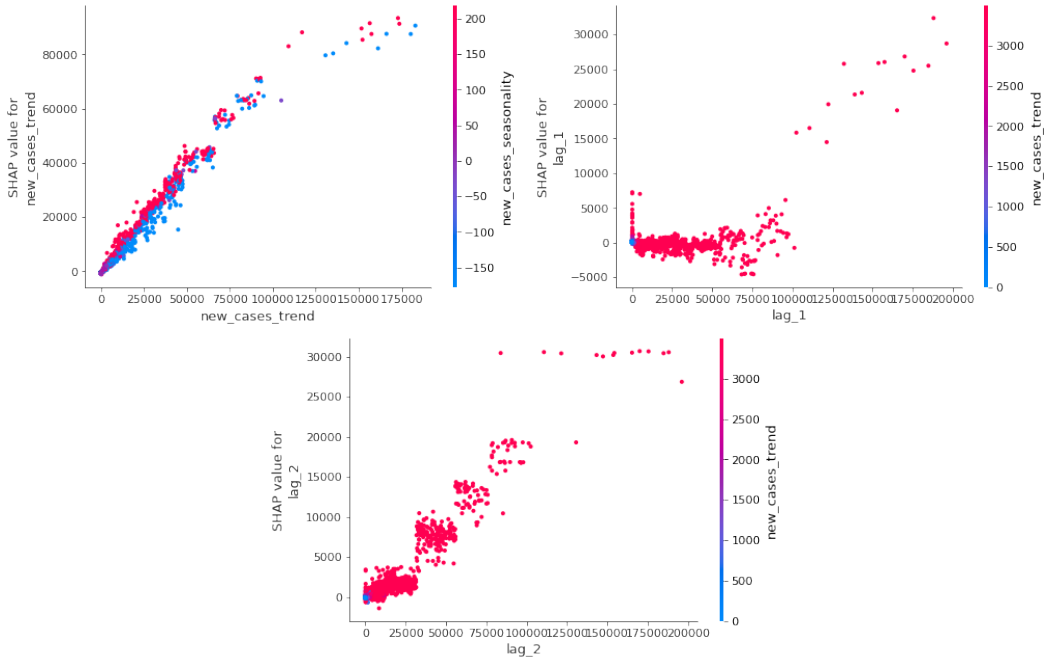
Variable Importance Plot

This simplified version can show us the most dense correlation, for example, new_case_trend in the non simplified version of this plot, shows a high and low correlation, however in the simplified version it signifies that the low dominates the high one. Same is also applied to other top features.

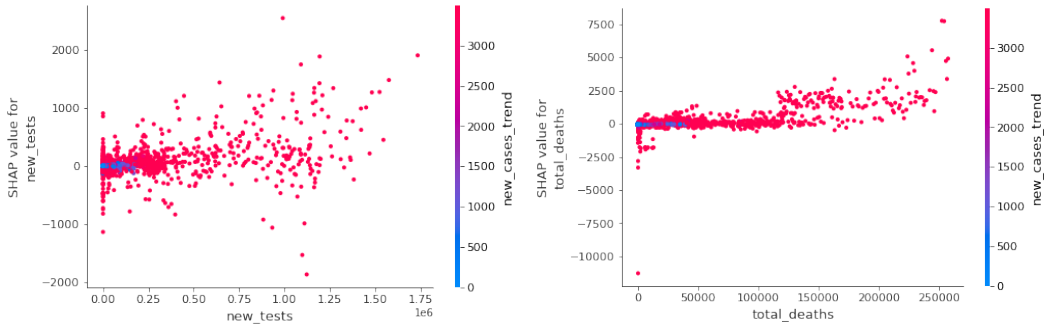
SHAP Dependence Plot

The dependence plot shows the marginal effect one or two features have on the predicted variable of machine learning model. This plot helps us indicating the relationship between the target and a feature and it could be linear, monotonic or more complex.

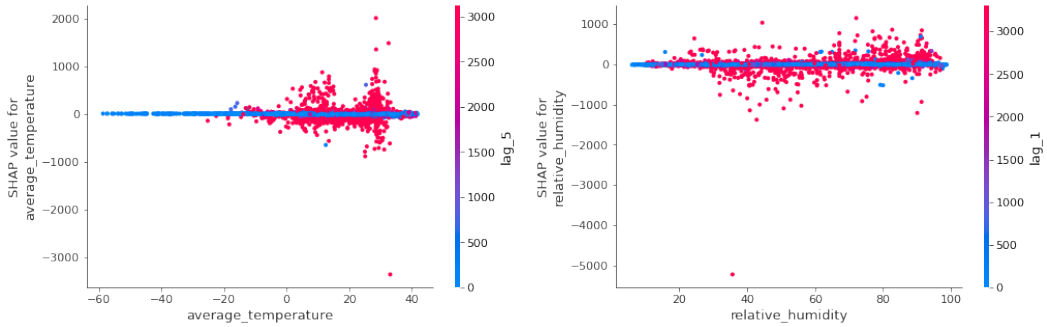
Below we show different plots of the top features, more plots are provided in the code.



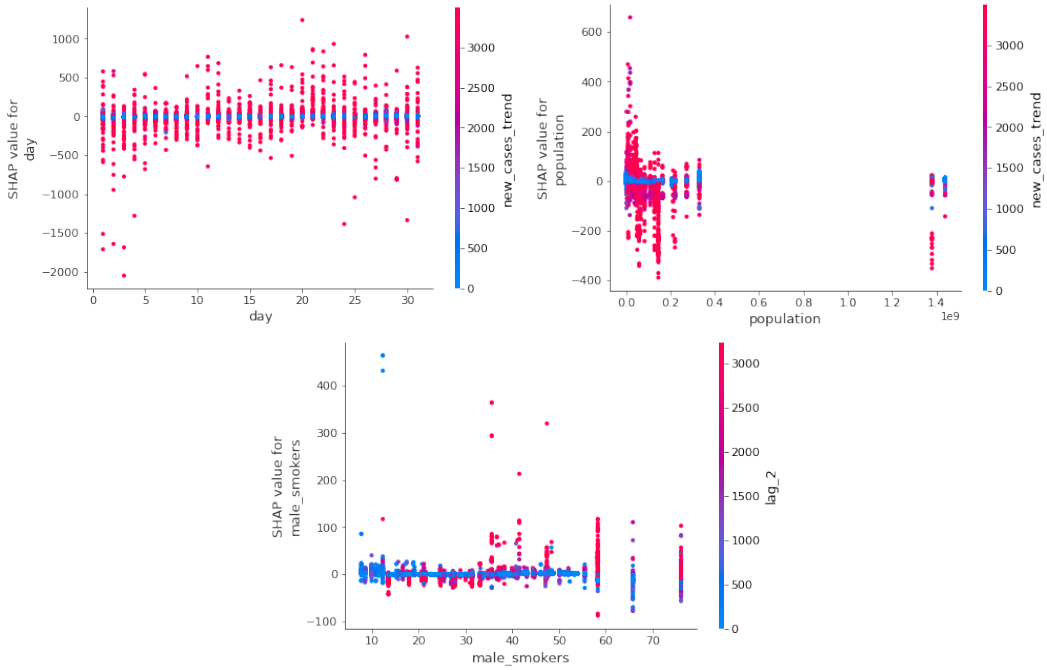
new_cases_trend, lag_1 and lag_2 have a positive linear correlation with the target variable. new_cases_trend interacts most with the new_cases_seasonality,and lag_1 and lag_2 with new_cases_trend (more positively i.e. red); the derivation of trends, lags and seasonality from the new cases can justify this interaction.



The new_tests and total_deaths graphs show a positive steady correlation with the target variable at the first range of the values, however as the new_tests number increases it shows a positive linear correlation with the target variable. Both features interacts most with the new_cases_trends, this can be explained that new_cases reported are related with the total number of tests performed per day, and as the new_cases rate increases, the possibility of death rate increases.



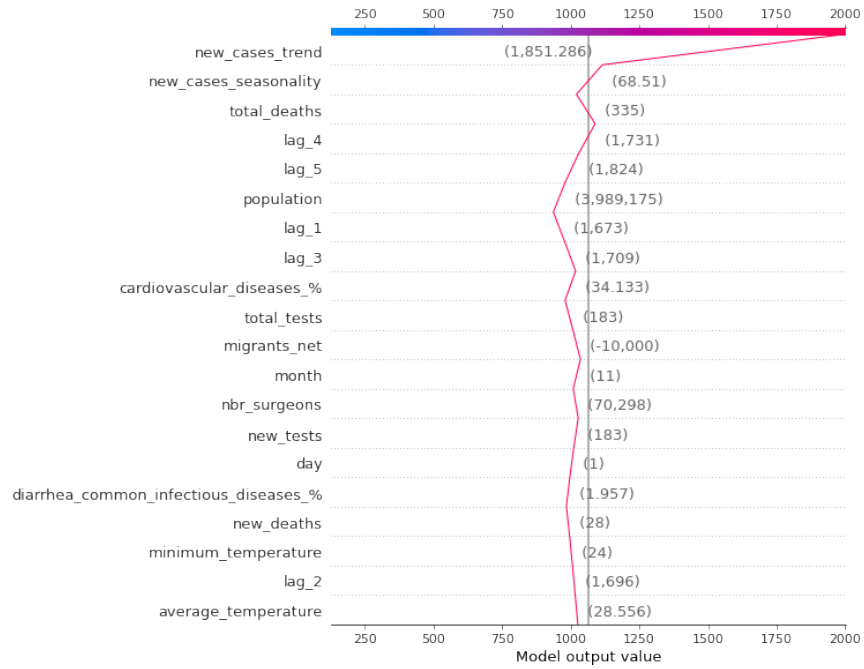
Although average_temperature and relative_humidity show on the top features that could have high predictive impact on the new_cases in the marginal interpretation, the dependence plots of the both features (as shown above) show a positive monotonic correlation and interacts with lag_5 and lag_1 correspondingly.



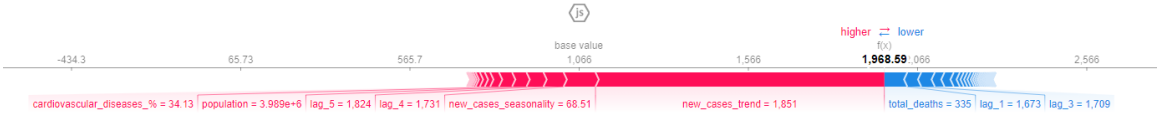
day, population, and male_smokers have a complex behavior on the target variable, we can say that the correlation is fluctuating between high and low. They interacts mostly with new_cases_trend and lag_2.

Local Interpretability Plots

Below we show sample of how the SHAP values can be applied to individual cases, we plotted the decision and force plot for specific rows.



Decision Plot of a sample data indexed at 1 of test dataset

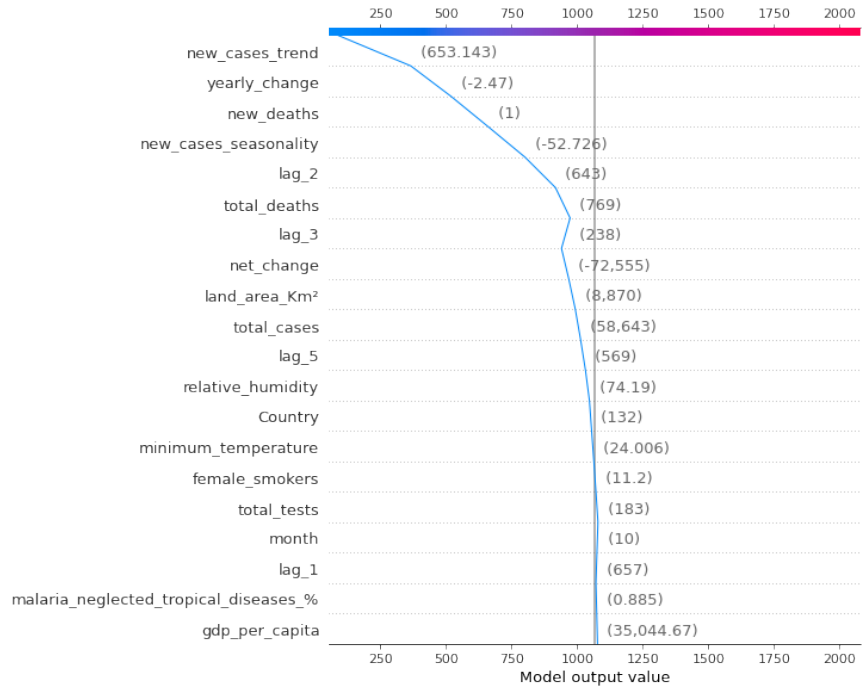


Force Plot of a sample data indexed at 1 of test dataset

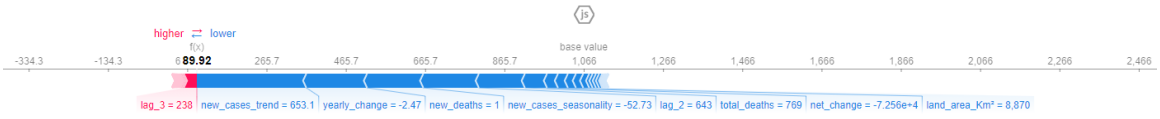
In the force plot, the red arrows represent feature effects (SHAP values) that drives the prediction value higher and blue arrows are those effects that drive the prediction value lower. new_cases_trend with value equals to 1851 increases the predicted value from 1100 to 1900 cases, and new_cases_seasonality with value equals 68 increases the predicted value from app. 800 to 1100 cases, both have high red arrows which positively affect the prediction of new_cases. Starting at the bottom of the decision plot, the prediction line shows how the SHAP values (i.e., the feature effects) accumulate from the base value to arrive at the model's predictive value at the top of the plot. Most of the features show a positive impact on the prediction till it reaches the prediction 1,968 new cases.

Below we show another sample from the test dataset and its predictive value at

index 0.



Decision Plot of a sample data indexed at 0 of test dataset



Force Plot of a sample data indexed at 0 of test dataset

new_cases_trend with value equals to 653 in this sample have a long blue arrows decreasing the new_cases prediction from about 300 cases to approximately 90 cases which negatively affect the prediction of new_cases. Looking back to the variable importance plot that shows the positive and negative relationships of top features, we can notice that the positive impact of new_cases_trend when its value is above 1000 whereas it shows negative impact when it is below. Starting at the bottom of the decision plot of this sample, the prediction line shows how the SHAP values accumulate from the base value to arrive at the model's predictive value at the top of the plot. Most of the features show a negative impact on the prediction till it reaches the prediction 89 new cases.

Conclusion

Researchers are combining efforts in studying the COVID-19 virus and finding effective solutions in containing the virus. In our project, we studied the worldwide spread of COVID-19 and explored the different factors that might cause individuals to be more susceptible to contracting the virus. Some of these factors were: reported new_cases, and new_deaths, weather per country, policies applied on different countries such as workplace closure, school closures, ... etc. , health index data , and country's population data. These factors helped us in predicting and forecasting the spread of COVID-19.
As data scientist, we started delving into our data and explore it more to be more knowledgeable of the problem we are tackling. First, we cleansed the data and changed the data type accordingly. Next, we checked for missing values and studied the possible techniques in imputing them. Also, we tackled erroneous values.
We explored our data and analyzed the features, we concluded the following:

- Denser countries have higher rate of interaction between citizens thus more COVID-19 cases.
- It is more effective on adults than the younger.
- In all the countries, a rise in total cases was preceded by less implementation of safety measures, and then followed by increase in safety measures.
- There is no clear correlation between different weather data and the total number of cases.
- COVID-19 could be more effective on people with diseases than healthier ones.

We converted our time series forecasting problem into supervised learning problem by computing the lag features , seasonality, and trend per country and add them to the data as new features.

We then modeled our regression problem using four different models XGBoost, decision tree, gradient boosting, and neural network. In this phase, we statistically evaluated the models and the results show that XGBoost, decision tree, gradient boosting perform the same and all are better than neural network model. We chose XGBoost model as our best model. The model suffered from high variance and low bias, increasing data size could help in increasing the bias and decreasing the variance

Our study was highly limited by the data provided because COVID-19 is a new

virus still in its early stages, and the numbers of daily reported new cases aren't accurate due to inaccuracy in data reporting, inability of some countries to do thorough testing. Moreover, since COVID-19 only started less than a year ago, there was not a huge amount of data to work with. Naturally, the size of the data will grow as COVID-19 continues to spread, however the quality of the data is dependent on the reporting from governments and their ability to do further testing.

SHAP

We used SHAP to promote confidence in our ML model. The results from SHAP showed a high dependency on new_cases_trend and seasonality and lags. This is an expected result, since the numbers from previous days is bound to affect new cases in the following days.

Back to SHAP global plots, we were also able to show that the temperature, and weather data in general, had no real visible effect on the number of new cases. This was seen previously when no correlation was found during exploratory data analysis. According to a scientific sources that are studying the COVID-19 virus, there is no way to predict how the virus responds in heat and humidity or cool and dry temperatures [1].This supports the reliability of our model results and its interpertibality.

However, even though population shows a positive correlation with the number of new cases, the SHAP results showed that the relationship fluctuates (positively and negatively). This could be due to the inaccuracy of new_cases being reported in some countries.

Thus, SHAP plots give us more confidence in explaining how our model is predicting the new_cases, some limitations is applied on our data and an insight from the end-user can help us analyzing the performance of the model more.

Research Project

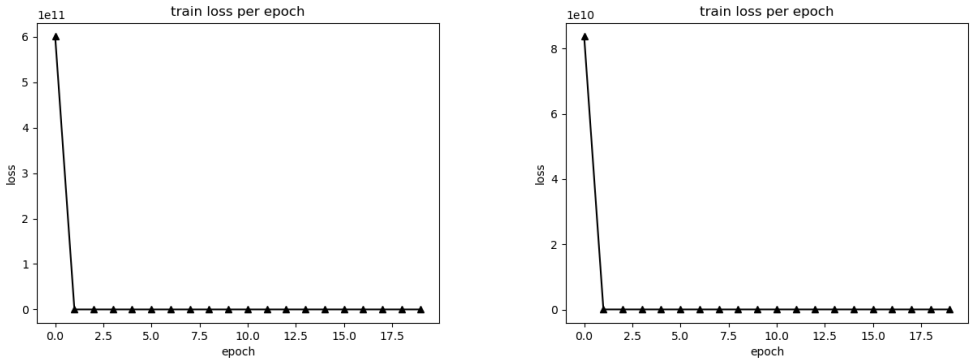
Uncertainty Estimation

In regular deep neural regression, we're learning a single parameter and using that parameter to explain how our data looks like. However, when we apply Bayesian neural networks, what we do is that rather that treating this parameter as a single point we treat it as a probability distribution over every parameter of the model. And since we start treating the input parameters as Gaussian distributions with a mean and a variance, the uncertainty of these parameters propagate to the final output, and thus make it easier to quantify.
In our project, instead of predicting a single value for our desired output, i.e. number of new COVID-19 cases, we return a probability distribution, and we get the mean of this probability and compare it the values predicted with our model.

Predicting the conditional distribution $p(y|x)$ evaluates the aleatoric uncertainty, the uncertainty that comes from the input, but it does not however cover the epistemic uncertainty, since information about the uncertainty in the model parameters is not taken into consideration. This often leads to highly confident predictions that are incorrect, especially for inputs x that are not covered by the training distribution, which might very likely be our case since COVID-19 is still a fairly new virus and thus some of our input variables can be unpredictable.

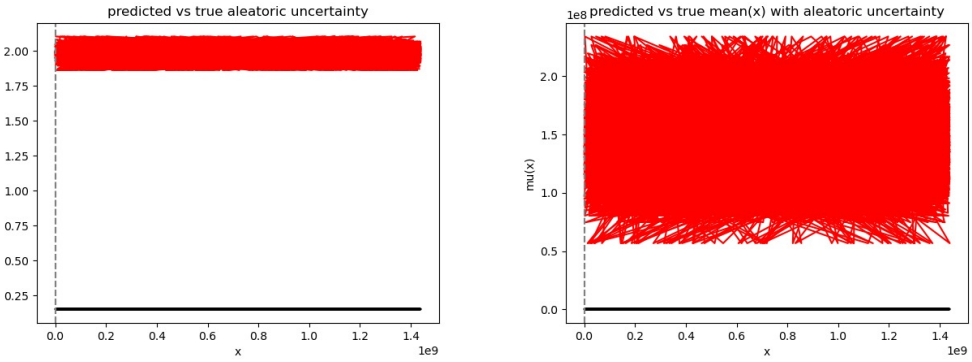
When we train the model several times, 256 times, and each time we have 20 epochs. Each epoch we get a certain amount of training loss. This loss is indicative of the quality of the prediction. This training loss reduces with epochs, and we notice that it plateaus around the 15^{th} epoch and settles at around 3.2×10^7. This is an MSE error.

The intuition behind the approach is that to capture epistemic uncertainty by training the model repeatedly and comparing the final result, i.e. from the last epoch, of the train loss and calculating the uncertainty from that.

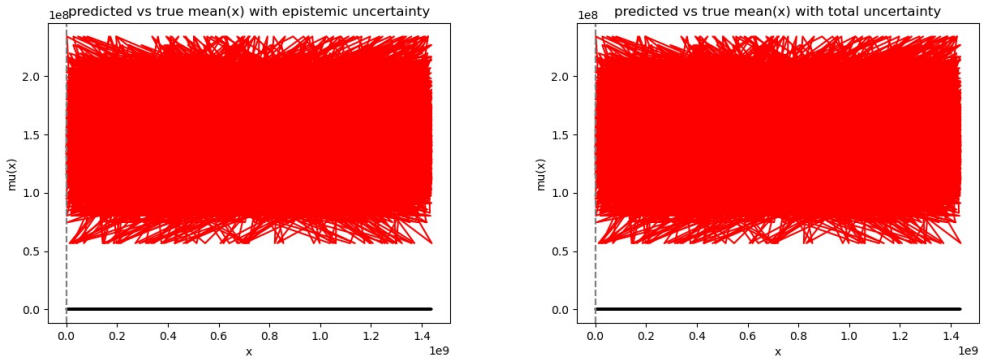


(a) (b)

Figure (a) is the first training of the model while figure (b) is the last one ($M = 256$). We see that the decrease of the train loss is consistent and it is so across the consequent training of the model.



The true aleatoric uncertainty from the data is given by the black line. The predicted uncertainty is given by the solid red color.
The mean of x which is shown as a straight line has no notion of uncertainty. The predicted mean(x) with aleatoric uncertainty gives a notion of the uncertainty in those values. This uncertainty obviously greatly affects the predicted value of our output. But this only covers aleatoric and not epistemic uncertainty.



(a) (b)

Figure (a) shows the predicted Epistemic uncertainty on the mean of x versus the mean(x) from the data. And similarly, it affects the predicted value of our output in a way that was not reflected by the simple mean of the input. Figure (b) shows both aleatoric and epistemic uncertainty.

The quality of the approximation of the uncertainties is dependent on the number M and the method employed. We used ensembling, which was shown to be best method for this estimation [2].

In the end, when designing machine learning models, the choice of whether to account for the uncertainty or not depends on the problem being tackled. In our project, the prediction of the number of new COVID-19 cases, it was important for us to take into consideration the uncertainty in the data because this information can have an effect on people's lives. However, we see here that the uncertainties obtained were very large, which is consistent with previous results, but it is a slightly larger than before. It makes it harder for us to say for sure how these uncertainties affect the prediction. There is inherently a large uncertainty in our data as the number of new cases varies wildly from country to country, and even in the same country from day to day.

Conformalized Quantile Regression

CQR is used to quantify the uncertainty in regression problems. This uncertainty can be captured using a prediction interval that has a lower and higher bound where the response variable occurs with high probability, given a miscoverage rate. The smaller the miscoverage rate, the less tolerance is allowed and the prediction interval has a higher chance of containing the response variable. Prediction interval provide valid coverage in finite samples without making strong distributional assumptions, and the shorter the interval is at each point in the input space the more deterministic

it is.

CQR inherits the finite sample and distribution free validity of the conformal prediction and the statistical efficiency of the quantile regression. It has proven to be adaptive to heteroscedasticity, and it has a thorough control of the miscoverage rate independent of the underlying regression problem.

In our project, we are using regression models to predict the daily number of COVID-19 new cases. Therefore, we used CQR to support our model with a level of confidence.

To do so, we used the code provided on our dataset and we had the following cases:

1. CQR with Random forest as the underlying quantile regression method using symmetric and asymmetric nonconformity score.

2. CQR with neural networks with and without rearrangement.

3. The same as 1 and 2 but with feature selection.

CQR with Random Forest

We ran CQR on our dataset with random forest as an underlying quantile regression algorithm using symmetric and asymmetric non-conformity scores. We set the miscoverage error to 0.1 and obtained the following results:

Method	Percentage in the range	Average Interval length
CQR Random Forests with symmetric score	99.533800	0.044116
CQR Random Forests with asymmetric score	94.955045	0.044116

CQR with random forest and symmetric nonconformity score achieved almost a complete coverage and had deterministic prediction intervals. Therefore, we can consider Random Forest as a confident model that can attain a valid coverage.

CQR with Neural Networks

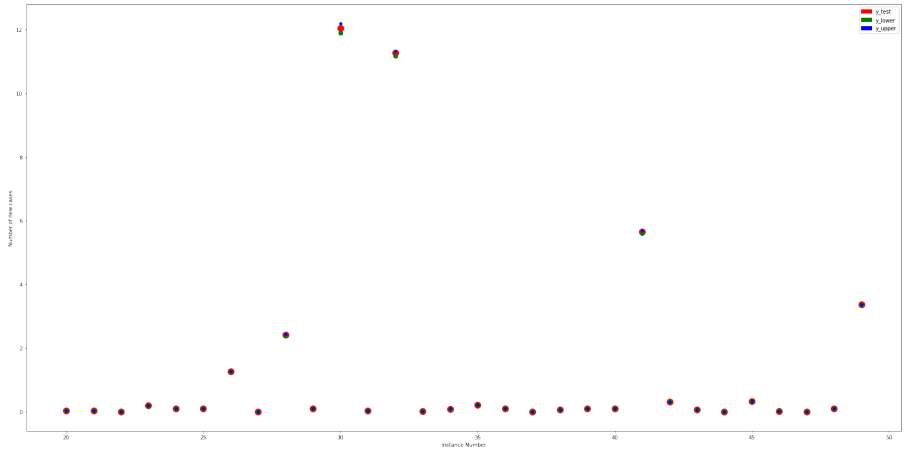
In this section, we ran the CQR with normal neural network and neural network with rearrangement, where the latter is used to solve the crossing quantiles problem. We obtained the following results:

Method	Percentage in the range	Average Interval length
CQR Neural Networks	90.517816	1.198820
CQR Neural Networks with rearrangement	91.516817	1.110525

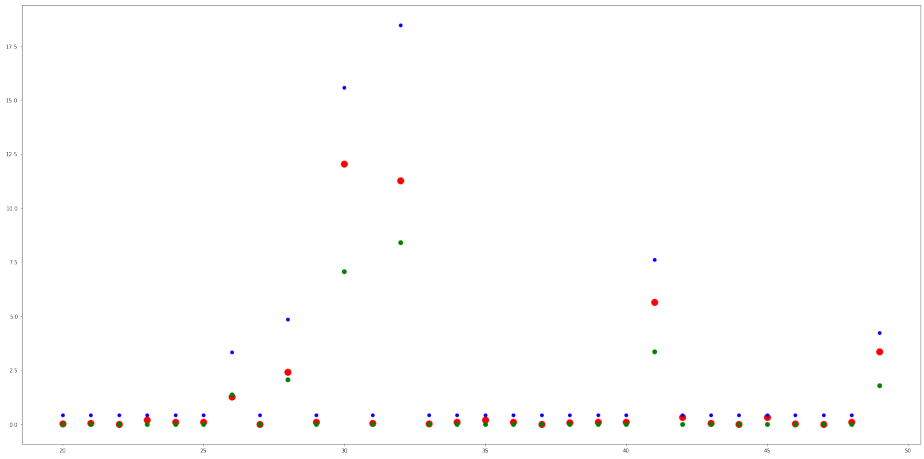
CQR with neural networks and with rearrangement achieved a slightly better coverage than that of normal neural network. In general, neural networks achieved a good coverage rate but not as good as that of random forests, and they had short interval length. Thus, random forests have a better confidence level than that of neural networks.
The below figure, gives a visualization of how upper and lower bounds are similar to the response variable. The x-axis represents the index of the instance taken, in this case we have taken instances from 20 till 50. The y-axis represents the actual number of new cases(red), upper prediction(green), and the lower prediction(blue). The coincidence of the dots indicates a valid coverage. It can be noticed that random forests have a better coverage than that of neural networks.

We have previously used XGboost, GradientBoostingRegressor, Decision Trees and neural networks to predict the number of COVID-19 new cases, and we have concluded from CQR that Random Forest have a high level of confidence so for future modeling we can use Random Forests as the regression model.

(a) CQR with Random Forests



(b) CQR with Neural Networks



CQR after Feature Selection

In this section, we explored whether feature selection affects the performance of CQR. So, we ran the same experiments as before but with feature selection and below are the key results obtained.

Method	Percentage in the range	Average Interval length
CQR Random Forests	99.367299	0.281615
CQR Neural Network	91.974692	1.679673

We can notice that feature selection had no significant effect on the coverage rate. However, it affected the interval length in a negative way. Thus, there is no need to use feature selection to obtain a better coverage.

Comparison with related work

We also compared CQR's behaviour, on our dataset, to other related work such as Split conformal, local conformal, and local conformal with median regression. We obtained the following results

Method	**Percentage in the range**	**Average Interval length**
CQR	99.533800	0.044116
Split Conformal	90.034	3.52
Local Conformal	89.918	22.417
Local Conformal with median regression	89.73	25.335

We can notice that CQR achieved the best coverage rate and interval length among the other related work.

References

[1] https://www.webmd.com/lung/coronavirus-heat

[2] Fredrik K. Gustafsson. *Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops: 318-319.