1 Introduction

The goal of this practical is to analyse number of cases and fatalities of COVID-19 between April 1 and April 30. The data includes a training and test set and external data which will be merged with the training and test set. Features will be explored and the data will be cleaned. Possible factors for transmission rate will be identified. Models will be fitted with the goal of predicting future number of cases and fatalities by area. The best model will then be uploaded to the Kaggle leaderboard where a score is calculated based on how well the model performs.

2 Data Exploration

The training set includes number of cases in each region and country by date. The test set has dates,regions and countries while the objective is to predict the number of cases and fatalities at future dates. The external data includes multiple variables for each country. Most of them are population or health related. The first step was to analyse the external data and pick out some features which could be interesting to analyse. The external data had many missing values and we picked interesting features with few missing values to avoid guessing or imputing a large part of the values. The features selected were population parameters and variables related to health and weather. All subjects discussed in the media as potential factors in affecting the spread of the virus. Before merging the external data with the original data set some cleaning had to be done. Even though we picked features with few missing values some missing values still had to be filled. Where it was possible the values were filled manually with real values but for the rest we filled the missing values with the mean value. Some inaccuracies exist in the data since not much external data was available for each region, in those cases the values for the country governing over the region were used instead. After merging the data sets together we analysed which features were most important in predicting the response which are future cases and fatalities. Using a random forest regressor the most important variables identified were region, days since data started being collected, population, gdp and median age. Surpisingly health statistics including number of hospital beds and health care expenditure were not important. Finally dummy variables were created for the categorical features and the final data set split into a training set and a validation set.

3 Model fitting and evaluation

The models will be evaluated using root mean squared error which will be abbreviated as RMSE. RMSE measures the difference between the predicted values and the true values. Specifically the square root of the average squared difference between the predicted value and the true value. The lower the error the closer the predictions are to the actual values. RMSE is also compared to the mean of the predictions which gives a better sense of prediction accuracy. The results can be seen in table 1 below.

	Cases RMSE	Mean	Fatalities RMSE	Mean
Linear Regression	7757	3.97	625	6.90
Random Forest	6053	3.10	436	4.81
Finetuned Random Forest	5522	2.83	416	4.59
LGBM	5801	2.97	501	5.53
Finetuned LGBM	8911	4.56	4072	44.94
ARIMA	1635	0.84	115	1.27

Table 1: Evaluation of predictions using RMSE and comparison to mean

The first model that was fitted was a linear regression. This model was used as the baseline model since it is simple and fast. It fits a straight line to the data which minimises the squared errors. After fitting the model on the training set we predicted the values on the validation set and evaluated the predictions using RMSE.

The random forest regressor was fitted next. It constructs multiple of decision trees on random subsets of the data. The output is the value that is the average value of values picked by the individual decision trees. It was picked since it is relatively fast and has performed well for past projects. The model was fitted on the training set first with 1000 estimators and 6 as the maximum depth. The predictions were an improvement from the linear regression. The model was then improved by finetuning the hyperparameters max depth and number of estimators. Grid search was used with four different values for maximum depth and number of estimators. The best random forest model returned had 1000 estimators and 8 as maximum depth. It performs better on the validation set than the previous random forest model as seen in table 1.

The next model used was light gradient boosting model or LGBM it was picked since it has performed well in many machine learning competitions. It is a boosting algorithm which behaves similar to other boosting algorithms but grows leaf-wise instead of depth-wise. The LGBM was trained with the default values, the predictions were not an improvement over the finetuned the random forest. Then it was finetuned by selecting different values for maximum depth, number of estimator, number of leaves and learning rate using grid search. The finetuned model performs worse on the validation set indicating an overfitted model.

The last model fitted, ARIMA, was fitted with a different approach than the previous models. The model uses past values of the response to predict future values of the response. Since we are predicting for each region we constructed seperate ARIMA models for each region using past values of each region. ARIMA models are popular models for time series data and were picked since we have time series data and using past values as a predictor seemed reasonable. The ARIMA model is a combination of an autoregressive model and a moving average model and has three parameters p,d and q. P represents the order of the autoregressive model or the number of lags of the response is used. Q represents the order of the moving average model or number of lags of the errors are used. D represents how often the model was differenced or the number of times the model has had past values subtracted. From the pmdarima package we used the auto.arima function which creates models with different combinations of the parameters p,d and q. The function returns the model with the lowest Akaike Information Criterion (AIC) score. The AIC is a score which is commonly used to pick between models. It penalises models with many parameters to avoid overfitting. This is done for all of the regions and predictions are obtained for each model. The RMSE score for the models is the lowest that has been obtained. This is our best model and the predictions were uploaded to kaggle where it returned a score of 0.89.

4 Conclusion

The last model fitted did well and I am reasonably happy with the results. The other models did not perform as well suggesting that those models could be improved or other models should be fitted. More time could be spent on data exploration and feature engineering. Exploring more potential features might help to improve predictions. Features related to tests per country and restriction efforts per country could be interesting to explore. Including feature interactions might also help. By combining features together and adding the best ones to the data set predictions could improve. Experimenting with more models, finetuning the previous models more or combining models could also yield improved results. Four models were tried in this project and many more exist. More time series based models which fit the data well could be fitted like vector autoregression or VAR which works similar as ARIMA except it allows the response to have lagged values of more than one feature.