Training a DeepAR model using Amazon SageMaker to forecast COVID-19 cases by country

1. Definition

Overview

Data analysis and predictions based on scenarios involving hypothetical factors have been important for devising policies to cope with the current pandemic, including measures such as national lockdowns. Being able to accurately forecast future transmission involves identifying underlying patterns and influences affecting the spread of the disease. The more that is known about what factors exactly have an impact on transmission - at both the level of unchanging characteristics of a country as well as measurable actions they have taken – the more well equipped states will be for tackling the problems they face. As this topic is having such a significant impact on the lives of such a large proportion of people around the world, I would personally like to undertake this project in an attempt to contribute to the scientific community's understanding of the crisis in any way possible.

Kaggle.com has launched the challenge of forecasting transmission, with the goal of helping to "identify factors affecting the transmission rate" by answering some of the questions posed by the World Health Organisation and the White House Office of Science and Technology Policy. These questions include the following:

- "What do we know about **non-pharmaceutical interventions**?"
- "What is known about **transmission, incubation and environmental stability**?"
- "What do we know about COVID-19 **risk factors**?"

Problem Statement

The problem being investigated as part of this project is defined as follows;

> *"To accurately forecast the number of reported COVID-19 related cases through training a DeepAR model, in order to help identify important factors affecting transmission rates."*

This problem involves gathering different data sets related to the virus, including the time series of cumulative cases, as well as a rounded set of metrics which can help to define the environmental and social risk factors and the non-pharmaceutical interventions made by governments. Once brought togther and appropriately preprocessed, this data can be used to train a DeepAR model to predict the future course of national case totals. Based on a defined accuracy metric outlined in the section below, the model can then be evaluated after each training sequence, and its hyperparameters subsequently updated to improve accuracy.

Once the accuracy is judged to be sufficiently accurate, the predictions will be re-run with certain factors 'turned off', in order to gain an idea of the model's reliance on that particular factor. For example, it is to be expected that a prediction of the number of cases would overshoot if the

presence of a lockdown were lifted (whereby its values were set to 0), but by how much would it overshoot? The calculation for this 'impact of factors' is also shown below.

Metrics

Shown below is an example of the calculation of accuracy at 80% confidence, where the confidence level indicates the range in which the target can fall to be classed as a correct prediction.

$$accuracy_{0.8} = \frac{x_{0.8}}{l}$$

x: element-wise sum of bool($C_{0.1}$ <= t <= $C_{0.9}$) along 1-D target and confidence bound arrays
where $C_{0.1}$ = 0.1 confidence bound, $C_{0.9}$ = 0.9 confidence bound, t = target

l: length of the target array

The average of this accuracy score, taken across all countries' series, will be taken to indicate the overall accuracy of the model. The hyperparameters of the model will be tuned according to this accuracy metric until it can't be significantly improved. Once a satisfactory accuracy has been obtained, the model will be re-run with certain factors nullified in turn to determine the positive/negative effect their presence has on the model's prediction. This will involve a separate calculation shown below.

$$impact = \frac{\Sigma_{i=1}^{n}\left(y_i^1 - y_i^0\right)}{l}$$

y_i^1: the i^{th} value of the new prediction
y_i^0: the i^{th} value of the previous prediction

This is similar to a mean square error calculation, but the absence of squaring meaning that the value retains its sign and allows for determining the direction as well as magnitude of impact. The units for this metric will be number of cases per million people per day. A higher positive value will be taken to indicate that a particular factor has a higher restraining influence on viral transmission, as its withdrawal causes the prediction value to increase.

2. Analysis

Data Exploration: Time Series

The data for the project comes from four CSV files. Three of these contain daily time series of cases and mortalities of viral outbreaks by country, one file for each of COVID-19, SARS and Ebola. The COVID figures [1] are constantly monitored and republished daily by the John Hopkins University Centre for Systems Science and Engineering, and made available for public use. The current total number of countries included at time of writing is 182, with each country's records beginning on 23rd January, meaning a total of 182 x 94 daily records like the one shown in fig. 1. This data set also contains provincial breakdowns for certain larger countries but these will be ignored for the

purposes of this project, as data on other topics such as demographic information is only readily found at a national level.

Id	Province_State	Country_Region	Date	ConfirmedCases	Fatalities
34377	Isle of Man	United Kingdom	2020-03-24	23	0

Figure 1. An example row of the COVID time series data

The data sets documenting the history of the SARS outbreak of 2003 [2] and the Ebola epidemic of 2014-2016 [3] are both sourced from Kaggle. These data sets, while only relating to a subset of the countries involved in the current pandemic, represent records of the entire timeline of previous viruses from start to finish and so are included in an attempt to help enrich the data. These data sets contain extra time series detailing the viruses such as number of recoveries and number of probable cases, however cumulative cases and fatalities are the only time series common across all of the data sets, with everything but cases to be ignored. The Ebola data runs for 573 days from August 2014 to March 2016, and the SARS data for 116 days from March to July.

Some portions in the SARS and Ebola time series are missing, and these will be interpolated linearly between the values at the start and end of the gap using Python. Also, many of the countries' values in this data set remain in single or low-double figures throughout and do not provide a significant trend suitable for analysis. For this reason, only Sierra Leone and Guinea's Ebola outbreaks and Singapore and China's SARS epidemics will be used for the purposes of this project.

As the time series describe nominal numbers of cases within a country, time series values will be normalized by population size, to be number of cases per million people, in order to give a better indication of the severity of the outbreak for a particular country. Each series will also have any inactive period at the beginning trimmed off, as this project is only concerned with the rate of development of an epidemic within a country, rather than trying to predict the starting point of the infection which is a much less predictable one-off event.

Data Exploration: Transmission Factors

In addition to the time series, a data set detailing environmental, transmission and risk factors has been collated from various sources as follows, in order to quantify the effect of a range of metrics on the model's prediction. Metrics are included for each country, under three headings: transmission/environmental; risk factors; and interventions. Table 1 below shows the metrics to be collected for the three categories, its numerical value and the intended source of the data. The metrics highlighted in orange have significantly greater measurable fluctuations in time and so will have fields relating to their changes in values at the points in time of these changes.

The number of monthly average temperature values for countries with SARS and Ebola outbreaks, as well as those which had early outbreaks of COVID-19 will be larger than most others, as their associated time series are longer. Some other factors, such as number of annual tourist arrivals and population density are unbalanced, with some extremely high outliers in both, for example France and Monaco respectively causing a skew which could cause a loss of information during training. In both these cases, more than 75% of the values are less than the mean, with the maximum value being more than ten times the mean. In these cases, the higher values will be clipped to a value close to double the mean.

Table 1. Table of transmission factors and data sources

Transmission/environment		
Metric	**Value**	**Source**
Border	Island (Y/N)	Google
	Landlocked (Y/N)	Google
Border activity	Annual International Tourist Arrivals	OurWorldInData
Altitude	Average elevation above sea level	Portland State University
Urbanization	Urban population percentage	World Bank
Pollution	Air pollution Index	Numbeo
Temperature	Average Celsius monthly temperature	World Bank
Population	Population area density	Wikipedia
Households	Average household size	Population Reference Bureau
Diversity	Ethnic fractionalisation	Wikipedia
	Linguistic fractionalisation	Wikipedia
	Religious fractionalisation	Wikipedia
Risk factors		
Metric	**Value**	**Source**
Wealth	GDP per Capita	International Monetary Fund
Safety of children	Infant mortality rate	Population Reference Bureau
Smoking	Annual cigarette consumption per capita	Wikipedia
Cannabis	Proportion of population using cannabis	Wikipedia
Obesity	Obesity rate	Central Intelligence Agency
Corruption	Corruption Perception Index	Transparency International
Crime	Crime rate per Capita	Numbeo
Cleanliness	Water and Sanitation Rating	Yale University
Interventions		
Metric	**Value**	**Source**
Lockdown	Proportion of population in lockdown	Wikipedia

Exploratory Visualisation

The plots of fig. 2 show a range of timelines from different countries and viruses, illustrating some of the variation in the data. Italy has a more severe outbreak than Suriname, and so has a smoother and more predictable curve than that of Suriname, which is yet to record many cases. Data such as Italy's is thankfully rare among the countries, however the lack of data series with such numbers means that the model can not yet be shown a broad range of mature epidemics from which to generalise, which will possibly stunt its performance. The Ebola and SARS data only begins part way through those outbreaks, and so their curve is already rising at the beginning of the series. These plots show how the virus numbers level off as an epidemic is eventually brought under control.

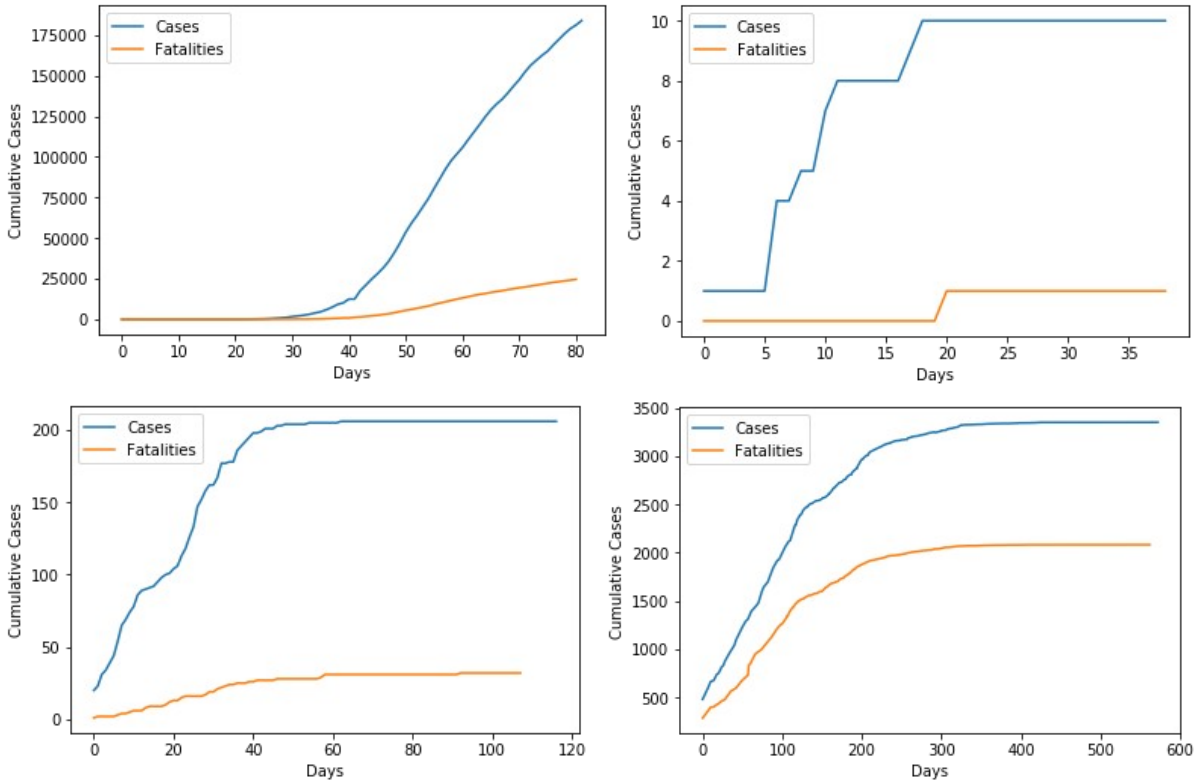


Figure 2. Clockwise from top left, the timelines of COVID-19 in Italy, COVID-19 in Suriname, Ebola in Guinea and SARS in Singapore

Algorithms and Techniques: PCA

This project will use PCA to reduce the dimensionality of the feature space amongst the transmission factors. Many of these will be highly correlated with each other, and the model will benefit from taking a smaller data set with fewer features which retain a high proportion of the original data variability. The proportion of original variability retained is represented by the follwing formula.

$$explained\ variance = \frac{\Sigma_n s_n^2}{\Sigma s^2}$$

In this equation, n represents the number of top components selected from the all those that can possibly be made. The fraction represents the ratio of the sum of the squared variances of these top components to the squared variances of all of the components. This allows selection of a small number of top components which capture most of the variance.

Algorithms and Techniques: DeepAR

The nature of the data in this project as well as its overall goal make it well suited to a DeepAR model, which is one of the built in Sagemaker models used for forecasting time series. It accepts time series as inputs, as well as an optional vector of features which allows for the inclusion of transmission factors. Although not so relevant for this project, DeepAR also has the capability of creating cyclical day and month time step features to help detect recurring patterns.

The model is trained through being shown random feature samples of a given ‘context length’, and its associated target of a given ‘prediction length’ which immediately follows it. The context and prediction lengths will be of significance in this project as it is important to know how far into the future the model is accurate for a given prediction window.

Benchmark

The standard of accuracy for the model in this project will be set by the DeepAR model trained in the Time Series Forecasting lesson of this nanodegree. This was trained on a large data set containing time series through years. Although the cyclical nature of the data was different to that of this project, it is a good example of the potential of the DeepAR algorithm and the kind of accuracy that can be achieved. Shown below in fig. 3 is an example prediction plot from the model. Its accuracy, based on the 3 prediction months and using the metric described in the first section, is 0.822. This will be the target accuracy for the forecasting model.

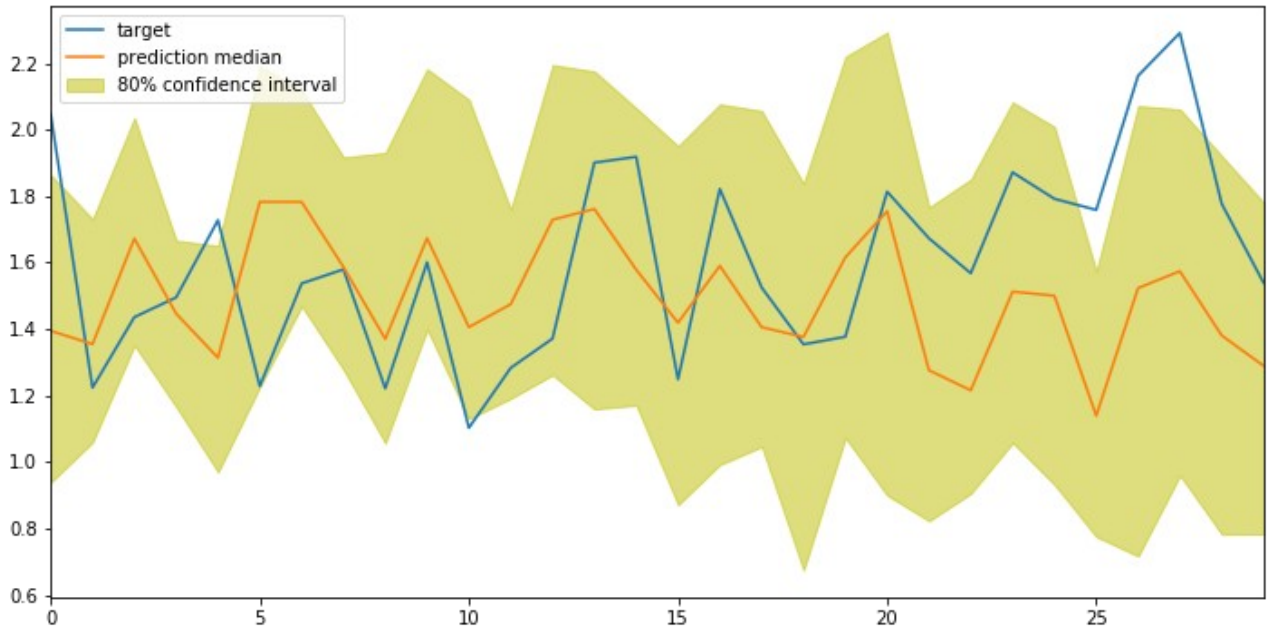


Figure 3. Prediction and target from energy data using a DeepAR model

3. Methodology

Data Processing: Time Series

The transmission data CSV file is imported into the notebook and read into a pandas daatframe, before being split in to two halves, one containing the static variables and the other containing those with changing values.

The virus cases time series are also read in and need to have their country names changed to match, for example ‘Viet Nam’ from the SARS data set needs to be replaced with ‘Vietnam’.

Then the interpolation needs to be performed on the SARS and Ebola data using a Python function. For this, it was necessary to instantiate a datetime object from the date field of each column, and

check whether this matched the date following the previous one. It involved creating a separate data set of interpolated values between the start and end dates of the gap, and appending this data set underneath the main one whilst iterating through it.

For the COVID-19 time series data, the individual provinces' data had to be merged into one row for each date, which involved iterating through the dataframe and finding all instances of a particular country on a particular date, and summing all values. Initially, a .groupby() method was tried, but this would have involved splitting the time series into cases and deaths, and possibly doing the same with the other data sets for consistency, so it was decided to implement a custom algorithm instead.

Following this preprocessing three data series were each converted to cases dicts with keys being countries and the values being their corresponding time series dicts in the JSON format required for the DeepAR input, e.g.

{ Afghanistan: {"start": "2020-01-22 00:00:00", "target": [0, 0, 0, 0, …..
…….

Irrelevant entries in each dict were removed, including all countries but those mentioned for the SARS and Ebola series, as well as the cruise ships in the COVID-19 data which are anomolous outbreaks with no available demographic data. Values of nationwide cases were nomalized using an iterative function to represent totals per million population.

Data Preprocessing: Transmission Factors

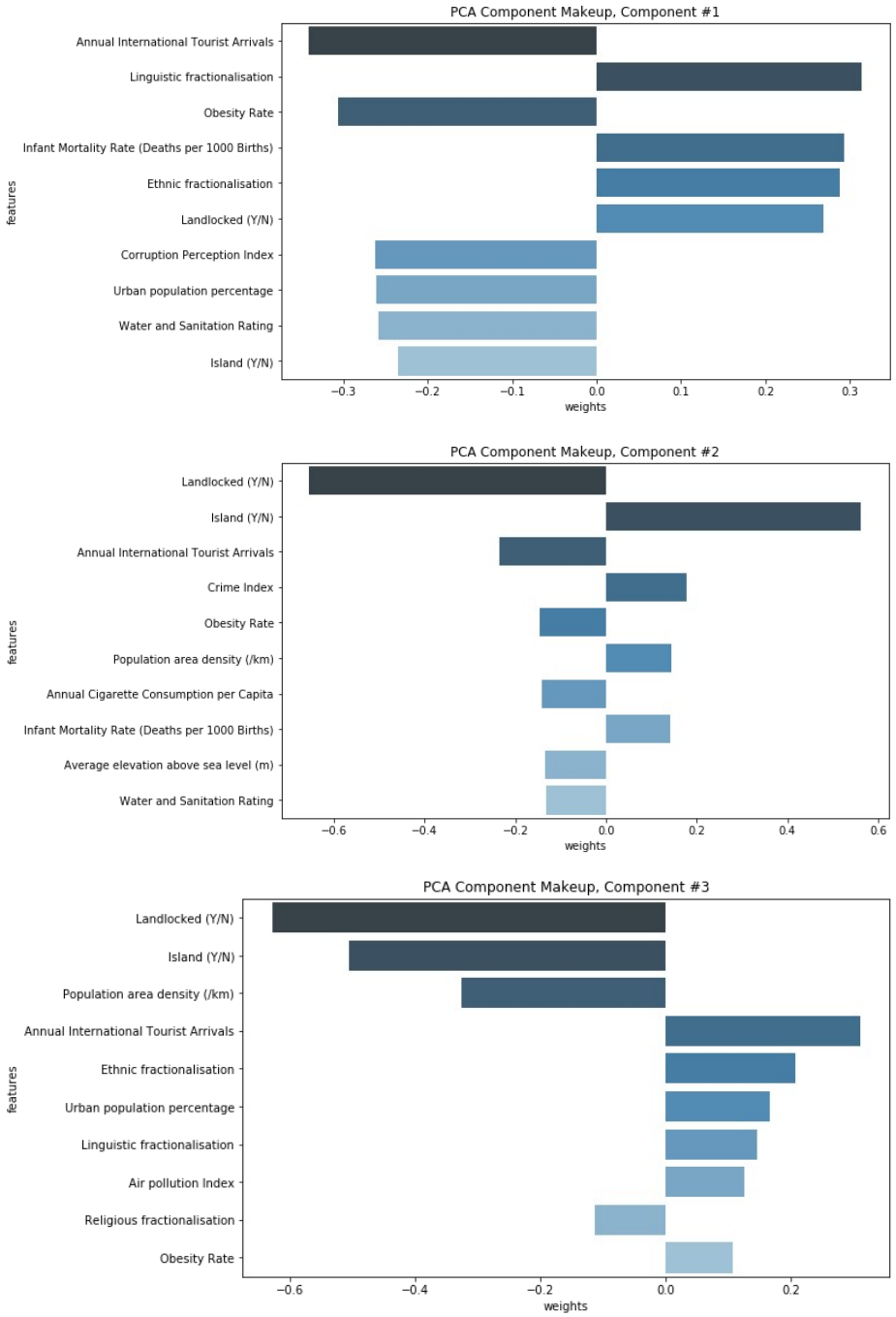
The time series data frame was visualised and its statistics analysed before the necessary clipping to reduce the effect of some outlying values. The country column was converted to be the row index and droped from the table, before all numerical values were normalized between zero and one using the sklearn MinMaxScaler object, making them more suitable for training. These values were then ready for a principal component analysis, detailed under the following heading.

Once the PCA had been conducted and a set of top components had been chosen and tabulated for each country, a dictionary corresponding to each time series was created, with countries in the dictionary containing their own dictionary of time series with corresponding length. These time series were copies of the static values repeating for the appropriate time window.

Separate dictionaries of time series were made in a similar manner for the temporal transmission factors, based on the dates in the corresponding cases data, were also made for each of the using Python scripts. All of the dicts for each country outbreak were then brought together into a single JSON object format for training the DeepAR, and uploaded to an S3 where they could be accessed during the training

Implementation: PCA

The Principal Component Analysis was carried out using a SageMaker PCA model with 18 specified components, which was trained using an ml.c4.xlarge instance and the resulting model saved to an S3 bucket. The resulting arrays were then loaded using MXNet and the previously mentioned equation was implemented to calculate the explained variance of the top number of components and to find out how many would be needed to capture at least 80% of the variability in the data. It was found that 6 components would be sufficient, and these are shown graphically below in fig. 4, visualised using the seaborn package.



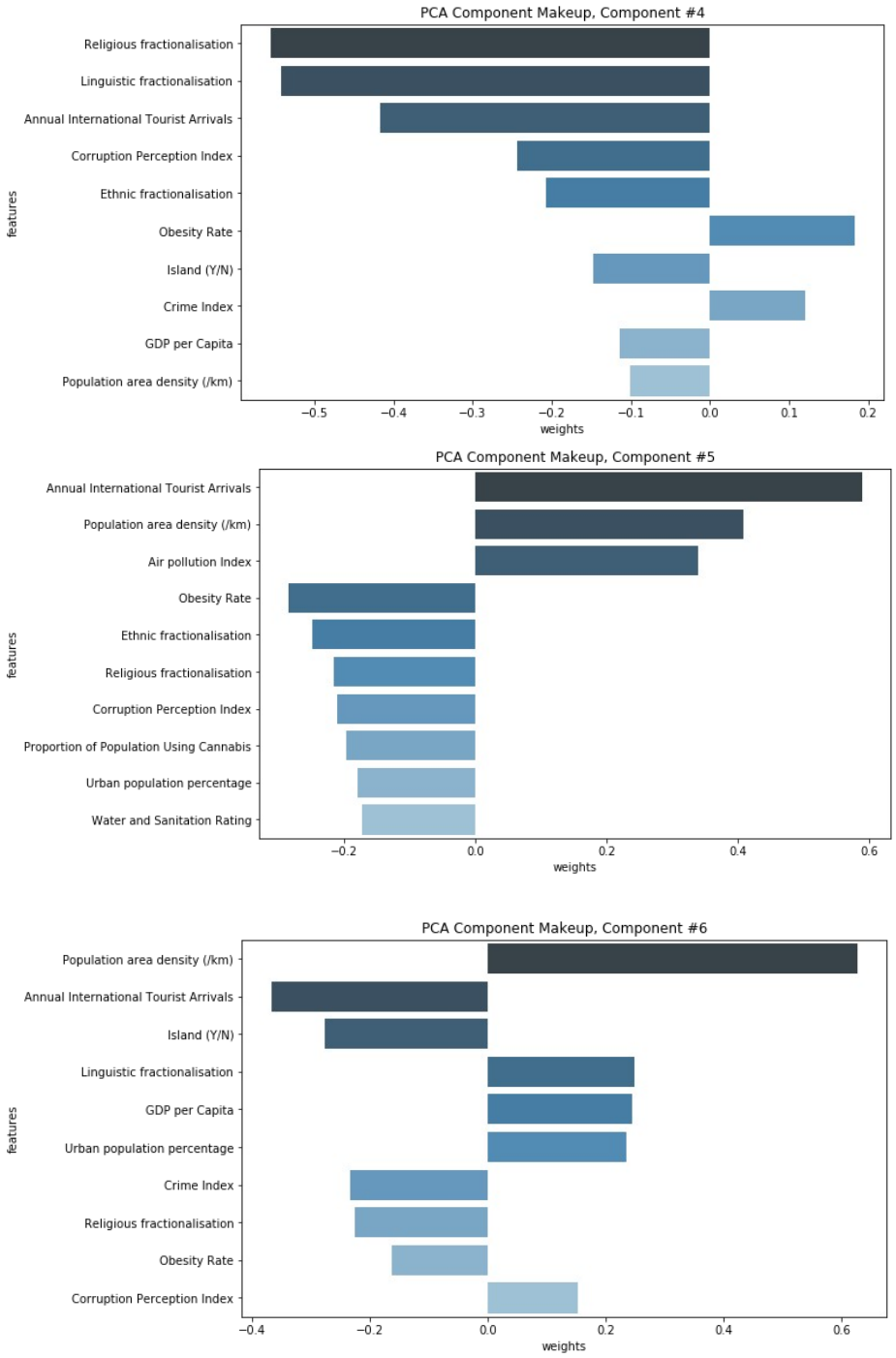


Figure 4. The 6 top principal components genrated through training the PCA object on the 19 static fields

Implementation: DeepAR

A Sagemaker DeepAR object was created and prepared for training on an ml.c4.xlarge instance. The input series for the model was the cases time series. The data would be provided to the training session in the form of a testing set, for validation, and a training set, almost identical to the testing set but with the last few values of the time series removed. The length of this missing data would be determined by the chosen prediction length of the model and would allow the DeepAR to avoid seeing information it would be asked to predict. The model takes a dynamic feature array comprised

of the series created from the transmission factor variables, with no need to shorten the length of the arrays for the training input.

Refinement

The model was trained starting from an initial set of hyperparameters, and experimentally tweaked to see what effect changing certain variables alone would have. The number of epochs was kept fairly low, around 25, both to save on training time and also because the loss often stopped improving at around or even before that point. The prediction length was set at 7 to make an adequate prediction whilst trying to ensure that not too many shorter time series were left out of training, and that predictions didn't become too inaccurate. The initial refinable values of interest were set as follows:

num_layers = 3
num_cells = 50
context_length = 7
batch_size = 128

These were tweaked and experimented with as the charts below in fig. 5 show, in order to try and find the set of hyperparameters with the lowest loss.

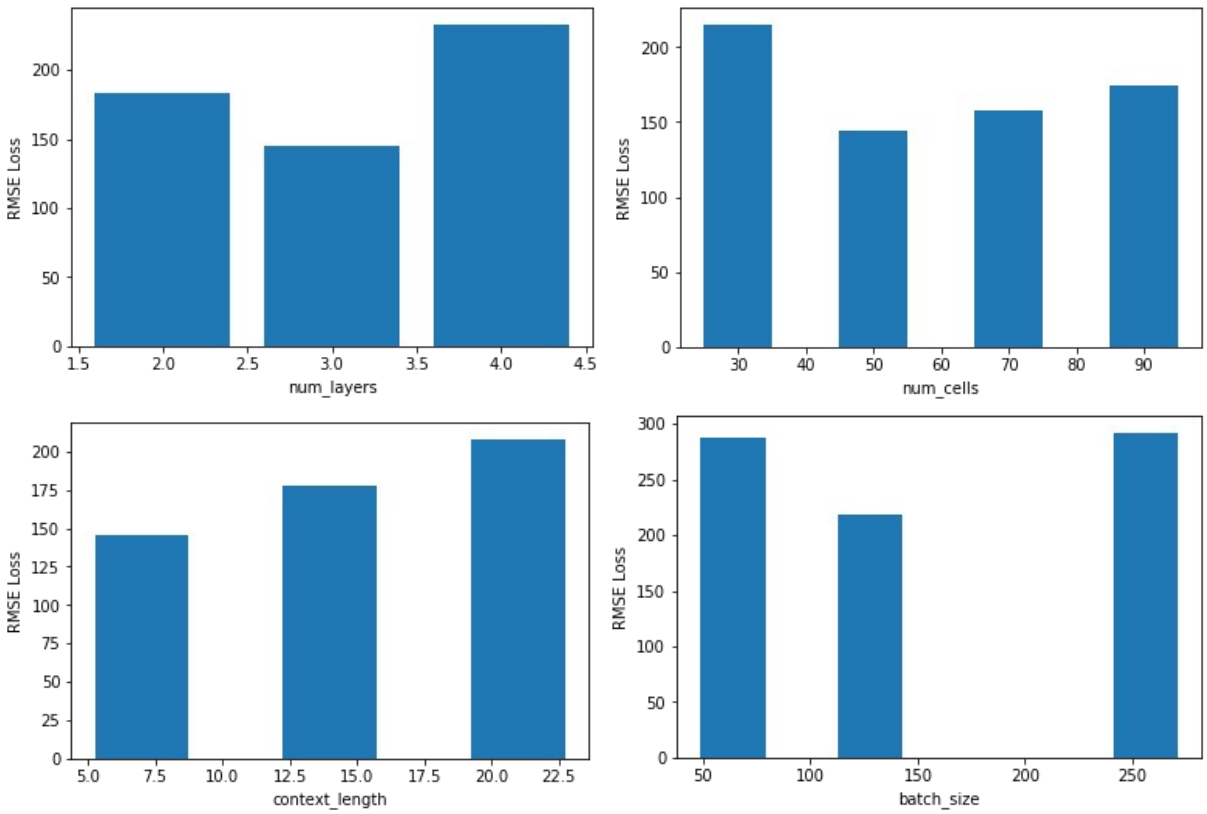


Figure 5. Clockwise from top left, the RMSE losses obtained from different values of: number of layers, number of cells, the context length (7, 14, 21) and the batch size (64, 128, 256)

The lowest RMSE loss was consistently seen in conjunction with the initial parameters, and so these values were chosen for use in the final model.

4. Results

Model Evaluation, Validation & Justification

Shown below in fig. 6 are a random selection of visualizations of the model's predictions against the target series. The blue series are the targets, the orange series is the prediction mean from 1000 generated samples and the highlighted area is the confidence interval.

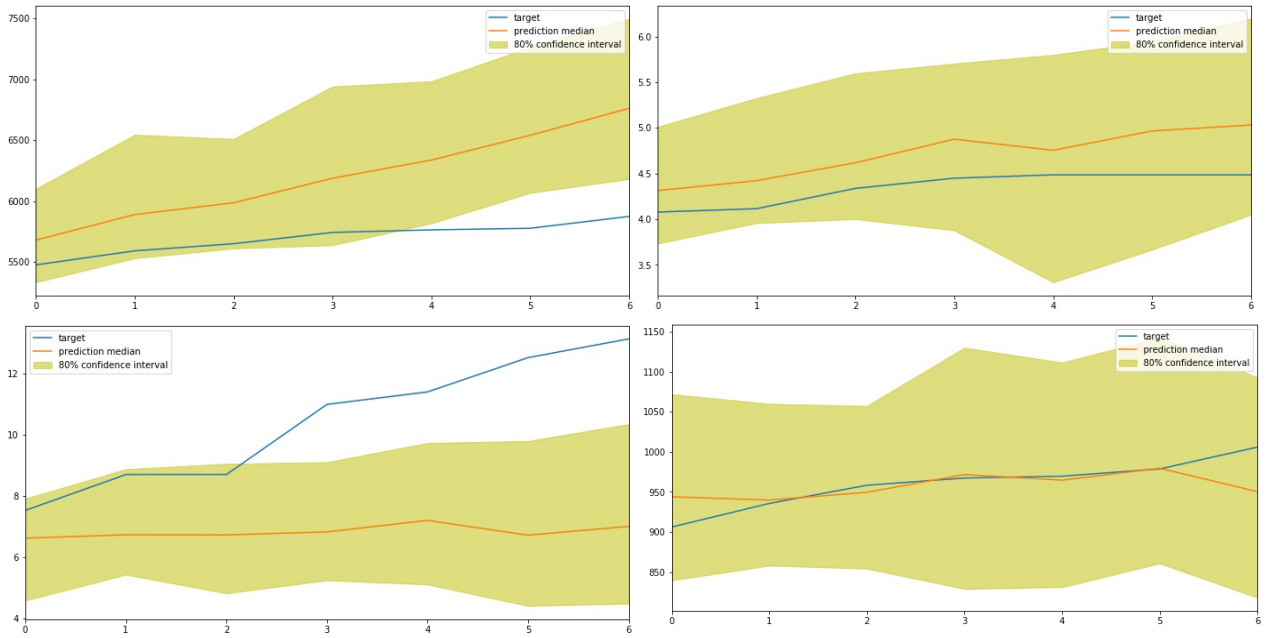


Figure 6. Plots of predictions and their confidence intervals with targets overlaid

The model achieved an accuracy, based on the 80% confidence metric, of 0.779. This is slightly lower than the benchmark of 0.822, however the data is not cyclical and cannot benefit from the in-built daily time step features in the same way that the benchmark model can.

Using this model, an impact score was calculated for each factor, by nullifying its influence and re-running the predictions. In the case of the static values, nullifying their effect was achieved through applying the lowest value of any country to all other countries, to see what the predicted effect would be were all countries low scoring in this metric. The impact scores for each principal component factor were calculated using the mean of 1000 prediction samples using the original and the nullified inputs. The results are shown below in table 2.

Table 2. Principal component impacts

Principal Component	x1	x2	x3	x4	x5	x6
Impact (reduction in new cases per million people per day)	13.2	18.7	18.2	43.4	61.4	28.1

The results indicate that for this model, the fifth principal component had the biggest effect on the output, with an impact score of 61.4. Predictions were therefore on average 61.4 cases per million per day higher with this factor “nullified” than with it left as it was. The impact score for x4 is also high at 43.4. Plots of the structure of both components can be seen in fig.4. Notably, both the top two most impactful principal components are negatively correlated with corruption perception index. Picking out example countries which score highly in some of the most impactful principal components, India being a good fit for component 5 and Israel for component 6, these countries either have low numbers of cases per million population or have seen their rate of daily case increases drop away recently.

The impact of the temperature and lockdown series on the results are shown below in table 3. The temperature seems to have larger positive impact than each of the principal components, indicating that higher temperatures bring down the number of cases significantly more using this model. The lockdown value is actually negative in this case, but the fact that it is close to zero illustrates the lack of effect that this seemed to have on the training of the model.

Table 3. Temporal series impacts

Temporal Series	Temperature	Lockdown
Impact	94.6	-10.1

Discussion

The accuracy for the model trained in this project fell short of that of the bench mark, but not by a significant amount. It had the additional difficulties of shorter time series and a lack of cyclical data. A rough idea of the impact of different factors relative to each other was found, although these values may change slightly with a retrained model. The sparse nature of the data made it difficult to optimise towards any kind of global minimum during training.

It is not easy to evaluate the use that a model such as this could have in the fight against a pandemic. For one, it is difficult to know whether one factor is caused by another or is simply correlated with it. Taking the impact of the weather as an example, it is difficult to know how much of the impact of the hotter weather on restraining cases is causal, and how much is due to the coincidence of governments making interventions at the same time as the Northern hemisphere moves towards Summer and temperatures in most countries begin to rise. In the model’s favour, however, there are no hugely unexpected results, and with more data accumulated over time it could be hoped to make more accurate forecasts.

It would be helpful for training a model such as a DeepAR if there were many more countries to use as data. A total of fewer than 200 different national time series is too small a sample size for the model to generalize to the different risk factors present within a country and tailor its predictions to these kinds of features. 20000 would be much more useful, however there are in reality a limited number of countries to choose from, and specific data such as water sanitation rating would be much harder to source at a provincial level. There is the added complication that many countries are unlikely to be recording all of their cases, whether this be because of technical or logistical difficulties or even political motivations. Because of this, patterns in the data may not be representative of the true transmission factors of the pandemic, and the model may even train to look for these flaws in the systems.

In conclusion, this project successfully forecasted COVID -19 cases with an accuracy not far below that of a chosen benchmark model. It also made an estimate of the types of country which have

more favourable factors for reducing the number of cases in the pandemic. Future work could be done into refining the model further, acquiring more data from different sources, and perhaps experimenting with algorithms other than the DeepAR.

Data References

[1] https://www.kaggle.com/c/covid19-global-forecasting-week-4/data

[2] https://www.kaggle.com/imdevskp/sars-outbreak-2003-complete-dataset

[3] https://www.kaggle.com/imdevskp/ebola-outbreak-20142016-complete-dataset