Modelling of COVID19 cases in London

Authors: *Chetan Khanna*, *Shashwat Oorjit Avasthi*, *Sanchit Krishna*, *Mayank Chaturvedi*, *Mudit Agarwal*

Abstract

The proposed work uses support vector regression and neural networks in an attempt to predict the number of active cases. The data is collected for the time period of 11th February, 2020 to 11th April,2020 (426 Days) for the city of London. The models have been developed to account for the previous 25 days of reported cases and deaths occurring due to Covid-19 along with other relevant features and then predict the number of active cases for the 26th day. The proposed methodology is based on prediction of values using a support vector regression model with Radial Basis Function as the kernel and a 2 layered neural network. The data has been split into train and test sets with test size 25% and training 75%. The model performance parameters are calculated as, root mean square error and regression score. We observed rmse of about 0.18 in predicting active cases using the SVR model and about 0.15 using 2 layers NN model.

Introduction

The spread of coronavirus disease 2019 (COVID-19) has become a global threat and the World Health Organization (WHO) declared COVID-19 a global pandemic on March 11, 2020. As of April 30, 2021, there were 151165770 confirmed cases and 3180150 deaths from COVID-19 worldwide.

The paper here discusses the proposed prediction model of COVID-19 spread in London using support vector regression, linear regression and neural networks implemented in python. The steps of the model are discussed in the methodology section with subsequent analysis. The results are shown and discussed and the overall conclusion is summarised in the Conclusion section.

Methodology

---

1. **Preparation of Dataset: For the forecast of active cases and deaths we chose 2 features to quantify the effects of lockdown and vaccination.**
    a. **Mobility: This data was extracted from the Apple website which uses data collected from Apple Maps**
    b. **Vaccination Doses: This feature was subdivided into first dose and second dose and was extracted from the official website for London data.**

    Apart from the above two features, we make use of the number of cases and deaths reported in the last 25 days as features to our model. This makes up the feature count to 53.

2. **Data Preprocessing:**
    a. **The available data of vaccine doses was available on a weekly basis. So we used linear extrapolation to get the number of doses given on a daily basis. (Except for the starting month for which the data was available in a 1 month gap).**
    b. **The data scrapped for different features are grouped in their respective folders in the "data" directory. Once all data was collected it was manually copied to a "data.csv" file which is what is used by the models and for generating the values for this report. This is done since the data was limited and combining it directly in excel was easier compared to programmatically doing it.**
    c. **The dataset is split for Training (75%) and Test (25%) using train_test_split() function imported from class model_selection of sklearn python library.The training and testing variables are saved for further evaluation.**
    d. **The training and testing variables of both X and y are standardized using StandardScaler() object imported from class preprocessing of sklearn python library.**

3. **Support Vector Regression**
    a. **Support vector regression is a popular choice for prediction and curve fitting for both linear and nonlinear regression types. SVR is based on the elements of Support vector machine (SVM), where support vectors are basically closer points towards the generated hyperplane in an n-dimensional feature space that distinctly segregates the data points about the hyperplane.**
    b. **The generalized equation for hyperplane may be represented as y = wX + b, where w is weights and b is the intercept at X = 0. The margin of tolerance is represented by epsilon ε. The SVR regression model is imported from SVM class of sklearn python library.The regressor is fit on the training dataset.**
    c. **The best model parameters from gridsearchCV are- 'C': 1.0, 'coef0': 0.0, 'degree': 2, 'epsilon': 0.01, 'kernel': 'rbf', 'tol': 0.001.**

4. **Neural Network:**
    a. **We have a multi layered neural network with ReLU activation functions.**
    b. **The parameters for the model are: learning rate = 1e-3, epochs = 40000, optimizer used was RMSProp**
    c. **These parameters as also the architecture of the neural network were chosen by trial and error**
    d. **The architecture was as follows:** Sequential(
        (0): Linear(in_features=53, out_features=80, bias=True)
        (1): ReLU()
        (2): Linear(in_features=80, out_features=80, bias=True)
        (3): ReLU()
        (4): Linear(in_features=80, out_features=1, bias=True)
)

Visualisation

We first plot the COVID cases and death curves for London.

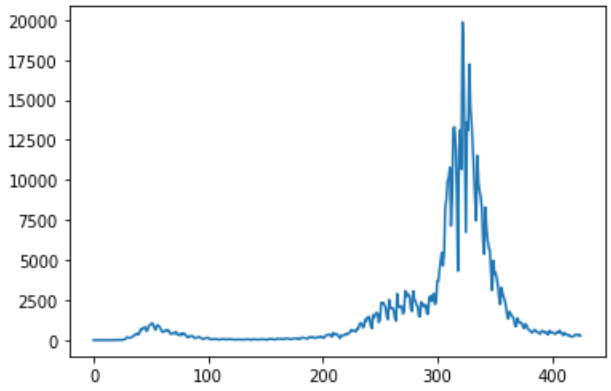

Figure 1: New cases vs Days

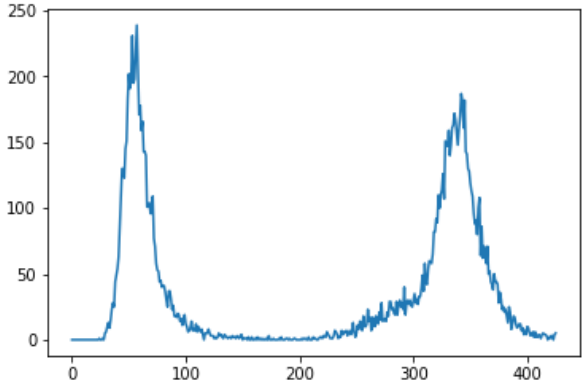

Figure 2: New deaths vs Days

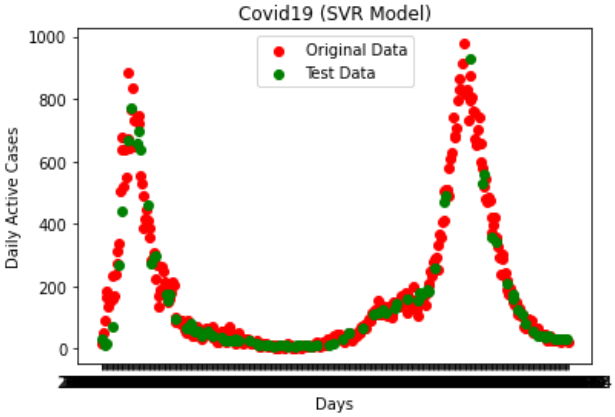

Figure 3: Original data vs Predictions from SVR model

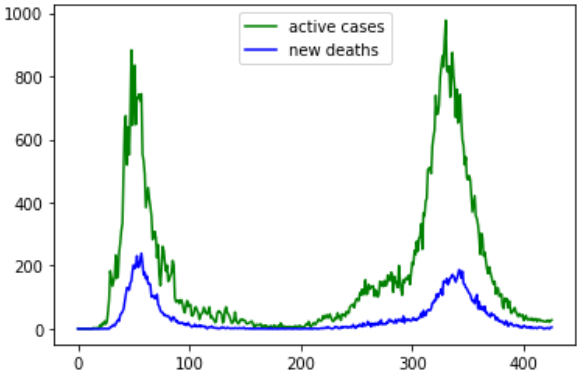

Figure 4: Active cases vs Reported Deaths

One of the reasons we selected "active cases" was because we speculated that there would be a strong relationship between active cases and future reported deaths as can also be seen from Figure 4.

Below is the graph for the loss function generated while training the NN (generated by wandb). The loss function used was RMSE. Along with it we also show Original vs Predicted Data on the same scale for our NN model (similar to the SVR model above).

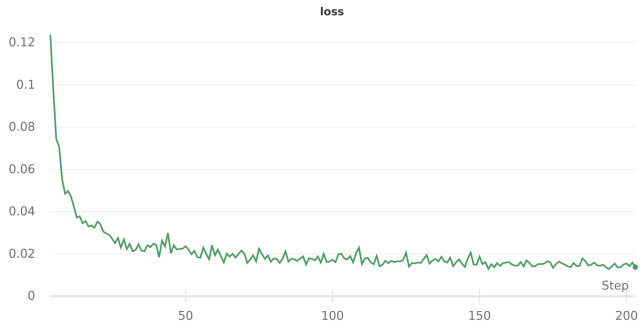

Figure 5: RMSE while training NN

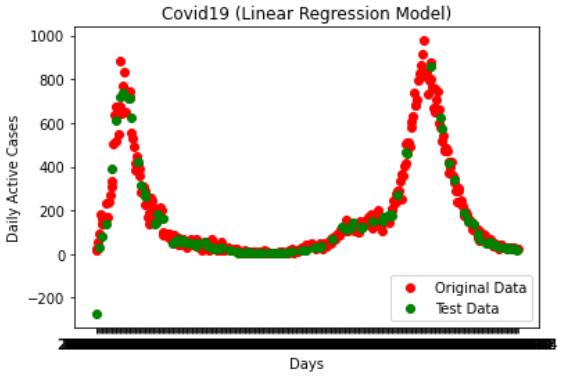

Figure 6: Original Data vs Prediction from NN

Model Performance Evaluation

The model performance parameters are then evaluated to check for the reliability in predicting the outcome. The root mean square error (RMSE) is calculated and shown in the table below.

Score	SVR model	NN model	Linear Regression Model
RMSE (test)	0.18	0.15	0.27

Predictions

Before splitting out the dataset into train and test sets, we held out the last 7 days of data to be used for prediction once the model was finalized. Here are the results for the same.

Date	Active cases (actual)	NN model prediction	SVR Model prediction	Linear Regression Model prediction
2021-04-05	23	26	31	15
2021-04-06	22	29	33	27
2021-04-07	26	28	34	22
2021-04-08	27	26	38	42

2021-04-09	**21**	**26**	**39**	**36**
2021-04-10	**25**	**22**	**36**	**20**
2021-04-11	**28**	**24**	**47**	**38**

It is to be noted that this data was never before shown to any of the models and was sliced off during the data preparation phase. We selected the last contiguous data to present the practical utility of our models.

Results and Discussion

The Support Vector Regressor(SVR) and Artificial Neural Networks(ANN) are both very popular and effective algorithms for regression. In the SVR case the rmse on the test sat was 0.18 while that in the ANN case was 0.15. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points, Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value. The threshold value is the distance between the hyperplane and boundary line. This makes the SVR particularly useful when used in the presence of noise, such as what one encounters in the present problem of pandemic spread forecast in a city, therefore the SVR model is robust for use in this case.

An ANN on the other hand, is a very good (universal) approximator, and it tries to find parameters, through optimization, that minimize the MSE loss. ANNs are prone to overfit, particularly in our case because of the comparatively low number of data points, therefore we chose a very shallow network, however their approximation abilities are brilliant, which is why the test mse was 0.15 for the ANN model. The ANN took as input a feature vector that contained data of the week before the date of interest and outputs xxxx on the date of interest.

On the note of importance of features for our model, we did a very naive experiment on the linear model: we plotted the relative importance of each feature on a simplified linear regression model.

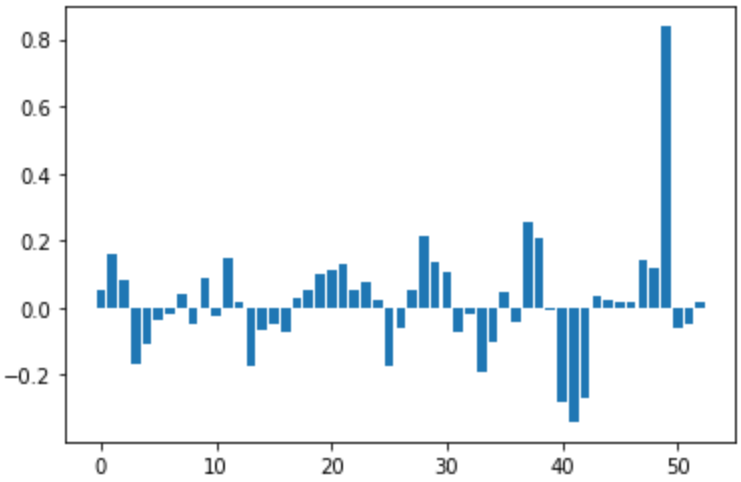

Figure 7. Relative importance of features

In Figure 7, the first 25 vectors are of reported cases, next 25 are of reported deaths and the last three are mobility, dose 1 and dose 2 respectively. From what we can see, the most prominent impact comes from the reported deaths on the previous day. Out of the last 3 features, dose 2 seems to have the most prominent impact on the regressor.

References

---

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7261465/
2. https://www.edureka.co/blog/covid-19-outbreak-prediction-using-machine-learning/
3. https://medium.com/swlh/covid-19-time-series-analysis-with-pandas-and-python-the-grim-rifr-b078e6a327ca
4. https://data.london.gov.uk/dataset/coronavirus--covid-19--cases
5. https://towardsdatascience.com/interactive-data-visualization-for-exploring-coronavirus-spreads-f33cabc64043
6. https://discuss.pytorch.org/
7. https://wandb.ai/
8. Statistics » COVID-19 Daily Deaths (england.nhs.uk)