**Predicting COVID-19’s trend**

*YOU Xinyu 1155110904*

**1. Introduction**

The pandemic of Coronavirus disease 2019 (Covid-19) becoming severe worldwide, especially in the United States (the country has the most registered cases at the time this study was done). Over 38millions of confirmed Covid-19 affection cases, and a million dead, it explained how worst the disease affects the world. Moreover, the rising number of infections the pandemic breaks out repeatedly in particular districts address the importance of prevention is crucial. Refer to the successful experience of taking control of the pandemic in some districts, for instance, Taiwan, Singapore, and some Asian countries, these prompt the importance of prevention cannot be ignored. Therefore, predictions on Covid-19's trend play an important role in assisting both the government and citizens to get prepared for the probable up-coming waves of infection.

Thus, some computationally competent and realistic models should be constructed to forecast the possible number of infection cases daily in the future, to provide reference or advice for policymakers to prepare for possible measures as well as getting the healthcare systems prepared for accepting new patients.

This study focuses on identifying the best-fitted model among some representative machine learning methods as well as make predictions of the number of Covid-19 cases in the future base on the selected model. The objective of this study is to find the best-fitted model and apply it to forecast the future incidence of Covid-19 confirmed cases in the United States.

**2. Methodology**

***2.1 Dataset***

Global registered Covid-19 infection cases are collected from the website—Kaggle(https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset). By attaching the confirmed cases data set in the United State which containing daily registered numbers from 20th January 2020 to 23rd September 2020.

***2.2 ARIMA model***

An ARIMA (p, d, q) (Autoregressive Integrated Moving Average with orders p, d, q) model is a discrete-time linear equation with noise, of the form

where is an Autoregression model with order and parameter ,

represents a difference operator with the number of differencing and time lag operator , stands for Moving Average model with order , parameter and error term of the autoregressive model.

***2.3 Holt’s Linear Trend Method***

Holt (1957) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend):

Forecast equation:

Level equation:

Trend equation:

where denotes an estimate of the level of the series at time *t,* denotes an estimate of the trend (slope) of the series at time t, α is the smoothing parameter for the level, (where 0≤α≤1) and β\* is the smoothing parameter for the trend, (where 0≤β∗≤1).

As with simple exponential smoothing, the level equation here shows that ℓt is a weighted average of observation and the one-step-ahead training forecast for time t, here given by ℓt−1+bt−1. The trend equation shows that is a weighted average of the estimated trend at time t based on ℓt−ℓt−1 and bt−1, the previous estimate of the trend.

The forecast function is no longer flat but trending. The h-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value. Hence the forecasts are a linear function of h.

***2.4 Support Vector Regression (SVR)***

The detailed description of SVM is therein in several excellent works (Vapnik, 1995, Gunn, 1997). A regression consists of a training data as given: , so that ai is a vector of real independent variables and bi the corresponding scalar real dependent variable. The regression equation in the feature space can be approximated by:

where ***w*** defines the weight vector, c is a constant,  is the feature function, and the dot product therein.

Minimize the following equation:

Minimize: and,

In the first equation, the LHS term represents the empirical error and the term C gives a measure of the optimization between the empirical error and the model complexity given by the second term of the said equation. The second equation defines a loss function called ɛ-insensitive loss function (Vapnik et al., 1996). The optimization problem is converted into the dual problem by incorporating Lagrangian multiplier β and β\*. Only the non-zero coefficients, along with their input vectors, **a**i, are termed the support vectors. The final form comes out as follows:

With the help of kernel function K(xi, xj), the SVR function can be obtained as given below:

The term, c, is calculated by using the Karush–Kuhn–Tucker conditions.

**3. Result and Discussion**

In this study, the dataset was divided into two sets to test the forecast performance of time-series studies, which the training set consists of the records up to 232th day; the test set data formed by records from the last 14 days. The forecast result from ARIMA and Holt’s Linear Trend for next 14 days will compare with the test set for evaluating the Root Mean Square Error (RMSE). Then base on the RMSE, SVR will be implemented to distinguish which method performs better and accurate in prediction.

***3.1 ARIMA model forecast result***

By implementing auto.arima function in Rstudio, suggests ARIMA(2,1,2) is a suitable model. By fitting Arima (2,1,2) in forecasting the number of cases in the next 14 days, the prediction is sometimes at 95% confidence and at around 80% of confidence interval(Fig1).

The accuracy measure of the forecast to our test dataset as below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| 11581.7207 | 13178.379 | 11795.177 | 27.57388 | 28.16867 | 4.1972746 | 0.04907896 | 1.771526 |

The accuracy of Arima (2,1,2) is about 71.83%(100-MAPE\*100%) and the RMSE is 13178.39, which is used for further discussion with Holt’s Linear Trend method and SVR.

***3.2 Holt’s Linear method forecast result***

The Holt’s Linear method was used to fit the daily number of registered Covid-19 cases, we can observe that the forecast value tends to slightly lag behind the true value in the training dataset (Fig2). The smoothing parameters were calculated which .

These values tell that the estimate of the current value of the level is based mostly upon very recent observations in the time series, however, the Value of the slope b of the trend component is nearly weighted none based on the very recent observations. In Fig3, we can observe most of the testing data falls by 80% CI.

The accuracy measure of the forecast to our test dataset as below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| 7966.7265 | 9700.180 | 7966.727 | 18.05243 | 18.05243 | 2.834933 | 0.018607789 | 1.293406 |

The accuracy of Holt’s Linear trend is about 81.95%, with RMSE 9700.18 which is less than Arima(2,1,2) then Holt’s Linear trend is more preferred in this cases of forecasting the number of cases of Covid-19 infections.

***3.3 Support Vectors Regression prediction result***

We first roughly make a prediction model by fitting the number of cases against a date into the svm() function to form a SVR. Then by predicting the value of the model we observed the predicted value by SVR is relatively close to the true value in our dataset, but the predictions are not ideal around the peaks. (Fig4) Then the RMSE was calculated which .

Although the RMSE of the preliminary SVR is better than both ARIMA (2,1,2) and Holt’s exponential smoothing, tuning the SVR model to the optimal level is still desired, to make the most accurate and ideal prediction. By defaulted *svm* function in R considers maximum allowed error () to be 0.1. By model selecting or hyperparameter optimization, we will start by doing a grid search. It means we will train a lot of models for the different couples of  and cost and choose the best one.

As in Fig 5 and 6, we can see the number of support vectors and RMSE is proportional to value. This means to get a smaller RMSE the should be relatively small enough. Then we use the tune method to train models with  and  which means it will train 88 models. By observing the plot of the result of the grid search (Fig7), we can try another grid search in a narrower range with . As we zoomed-in inside the dark region we can see that there are several darker patches. From the graph (Fig 8) we can see that models with C between 300 to 500 and ϵ between 0.10 and 0.15 have fewer error.

Finally, by using Rstudio to assist us in selecting the best model, we find would be the best model. By visualizing our best model, and the preliminary model (Fig9), the best model fit better as well as closer to the actual value than the preliminary model. Besides, the RMSE of the best model is 4637.52 (RMSE of preliminary model= 6456.491).

**4. Conclusion**

In this study we have compared the RMSE of different prediction or forecasting methods, support vectors regression will be the best method for predictions in this case, according to the most accurate prediction (Fig9), we found that the spread of Covid-19 infection will start to rise from the late September in the United States. Base on the results and predictions, we can address there is a possibility of a third outbreak in late September in the USA. Therefore, everyone should play their role well in the pandemic, for instance, keep 1,5meters social distance, self-quarantine, or follow the guidelines provided by the government. With further research on Covid-19 released, humans can overcome the pandemic in the coming future.

**5. Reference**

Vapnik et al., 1996 V.N. Vapnik, S. Golowich, A.J. Smola

**Support vector method for function approximation, regression estimation and signal processing**

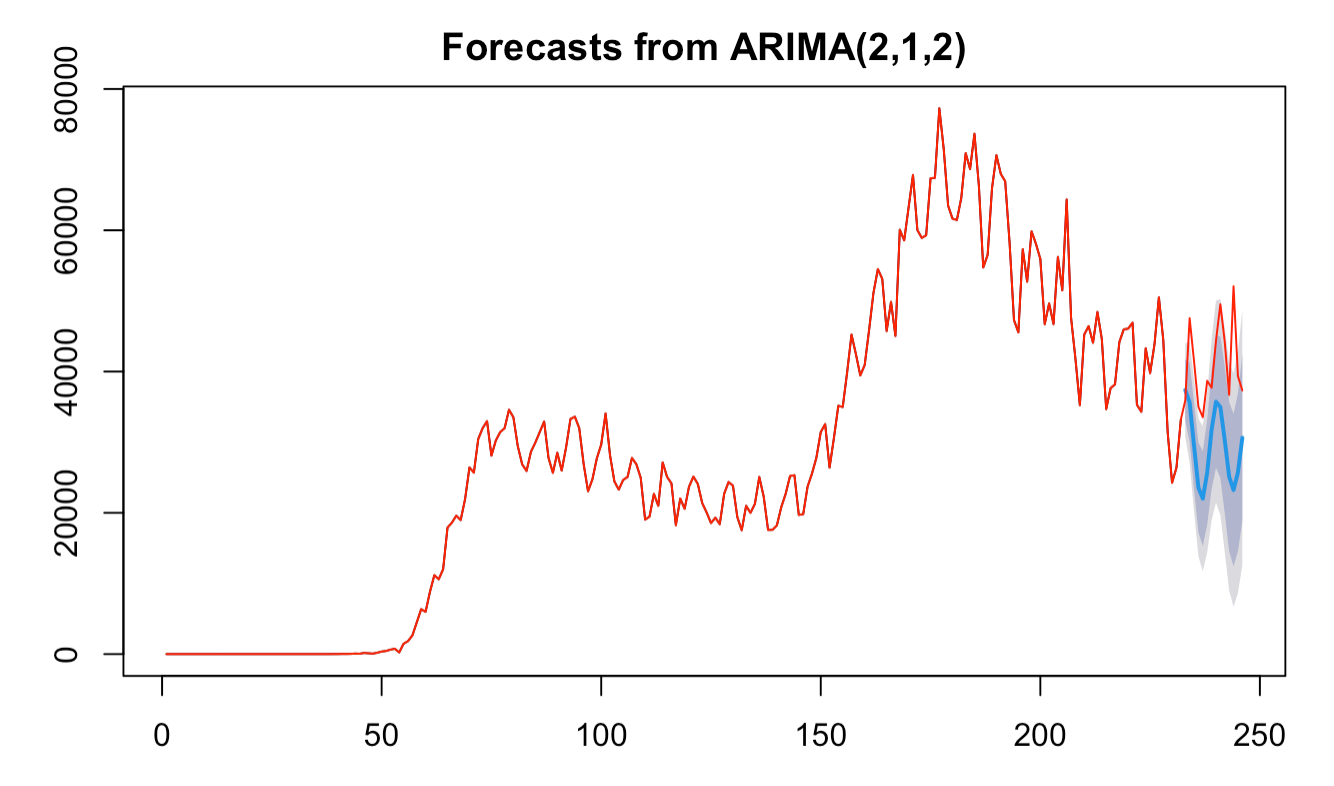
Adv. Neural Inform. Process. Syst., 9 (1996), pp. 281-287

**Appendix**

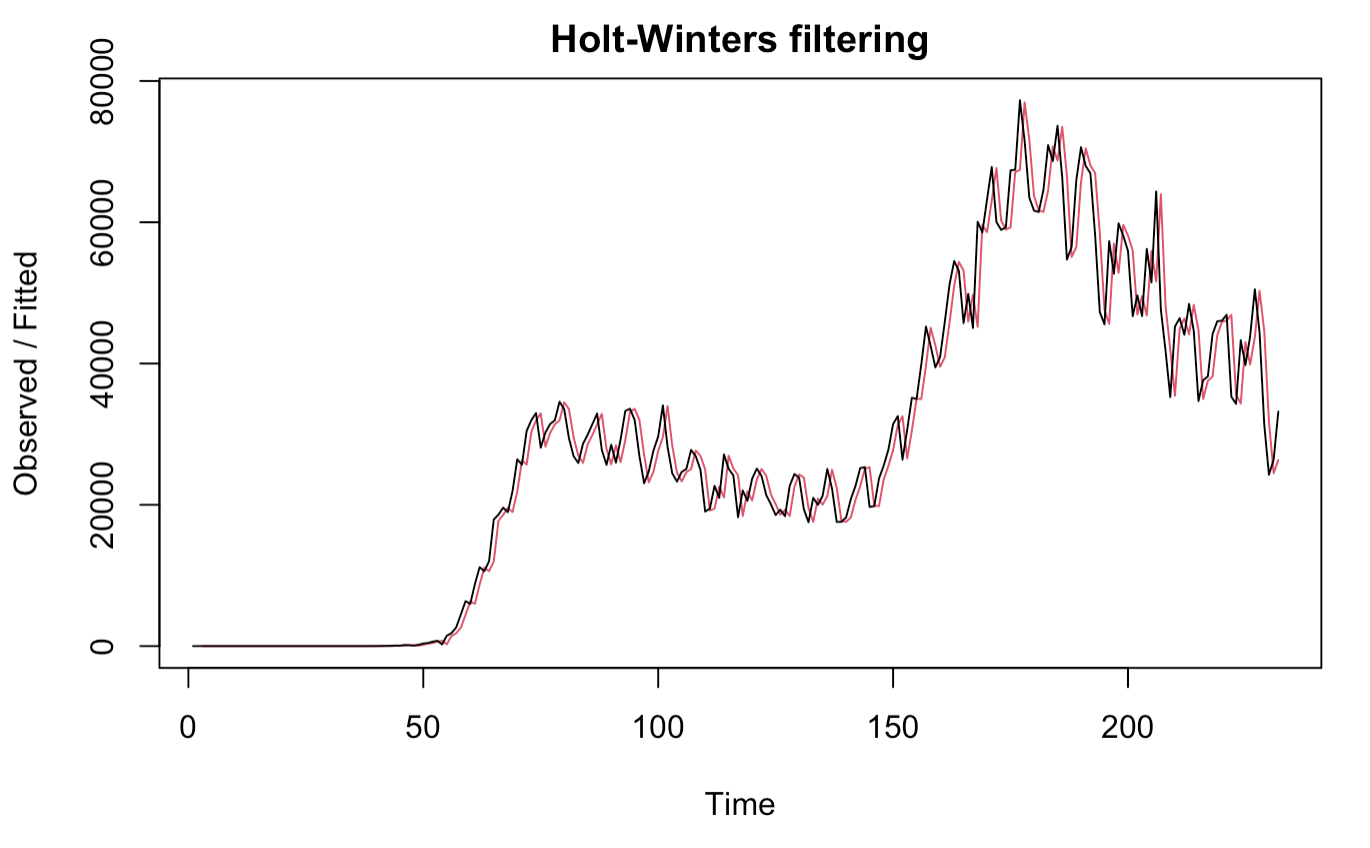
Abbreviation

(ME: Margin of error; RMSE: Root mean square error of fitted model; MAE: Mean absolute error; MPE: Mean posterior estimate; MAPE: Median absolute prediction error; MASE: Mean absolute scaled error; ACF: Aberrant crypt foci.)

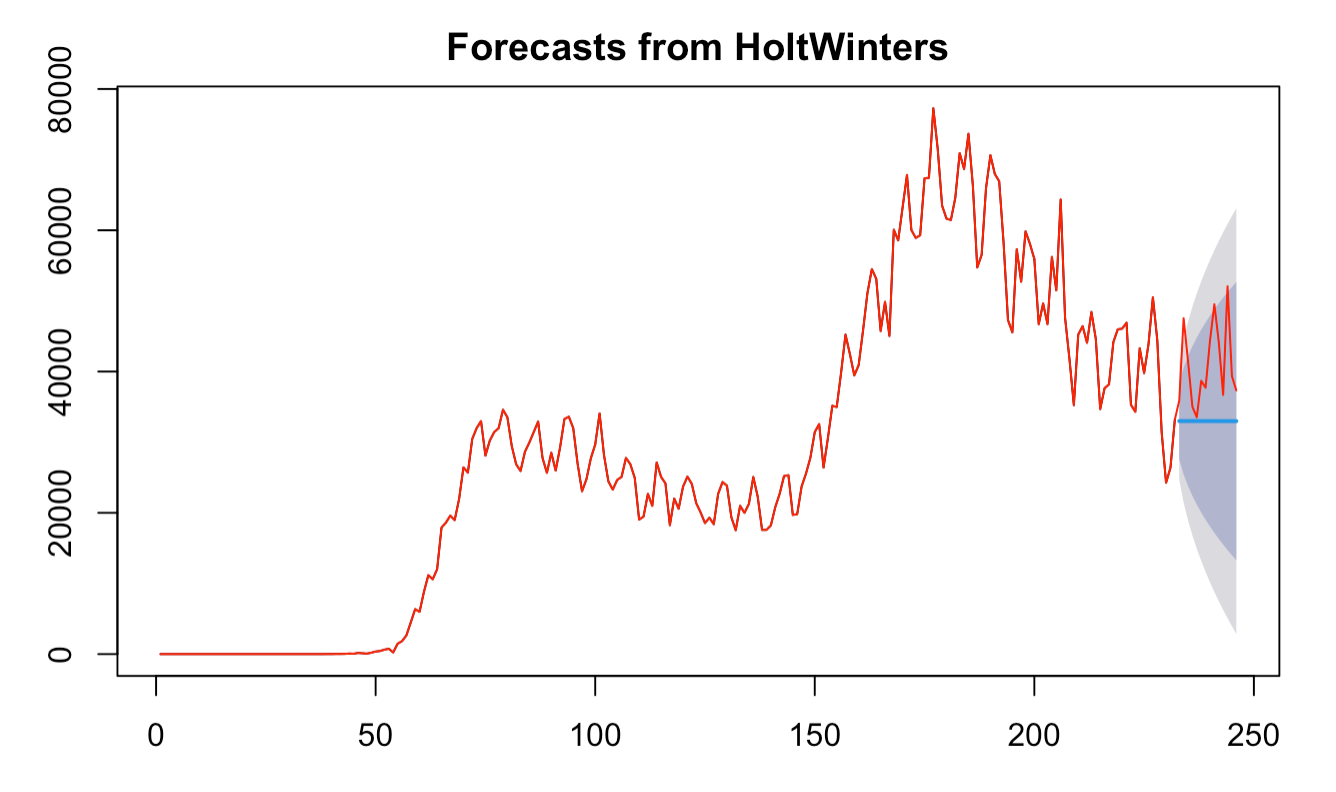
**plots**

Fig1

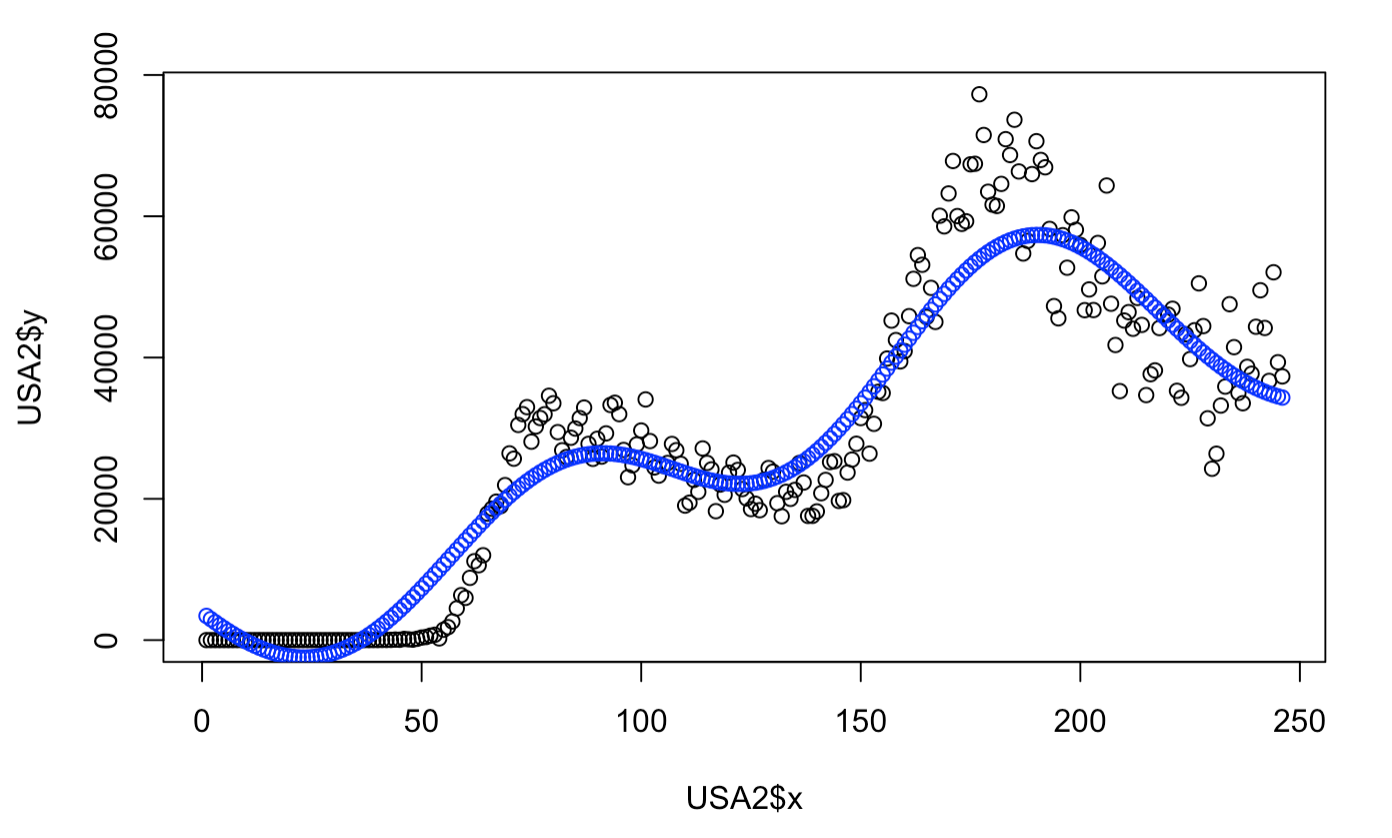
(red line : the actual data, blue line: the forecast result of next 14days, greyer zone : 80% CI, lighter Zone : 95%CI)

Fig2

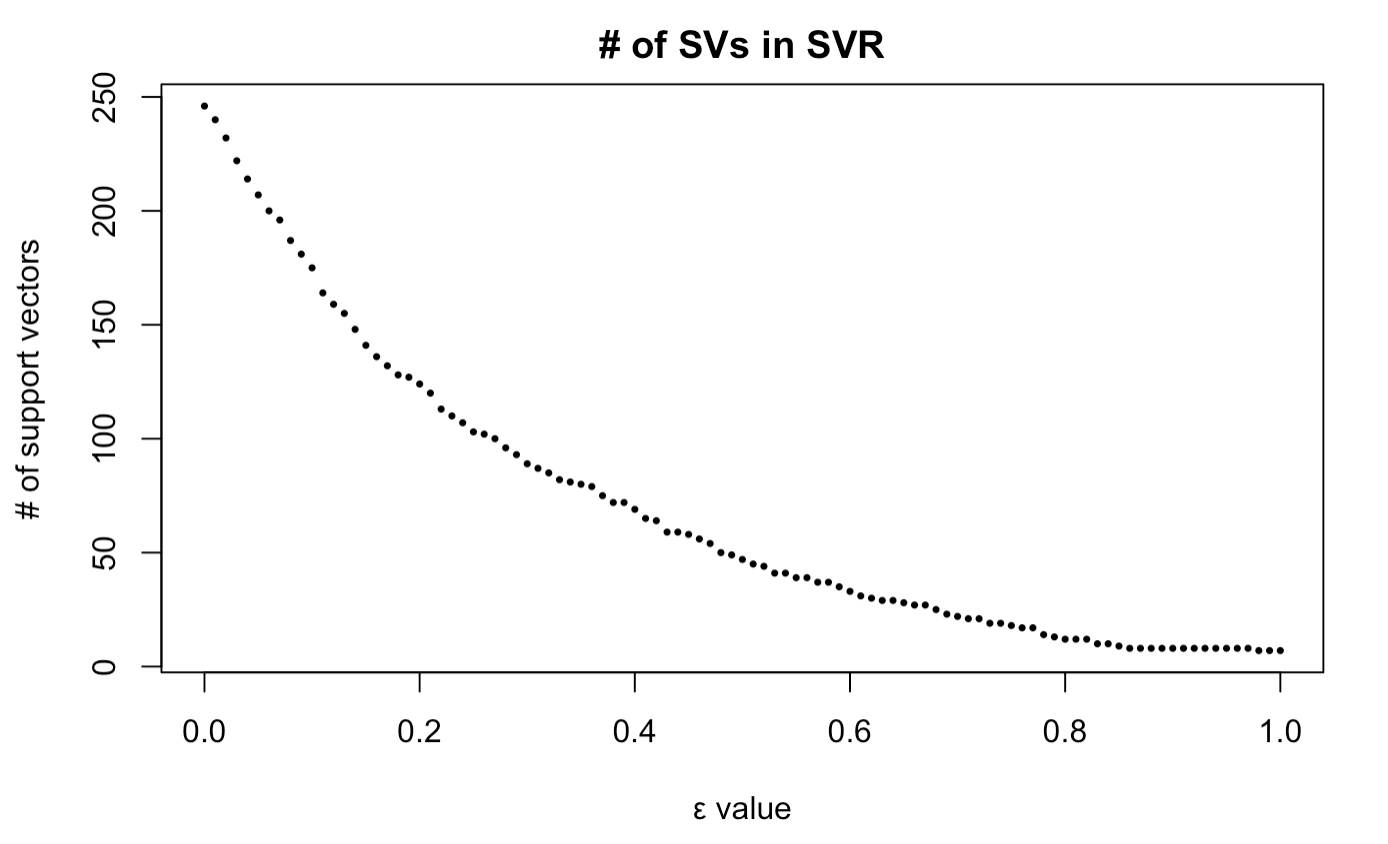
(black line = actual value , red line= forecast value)

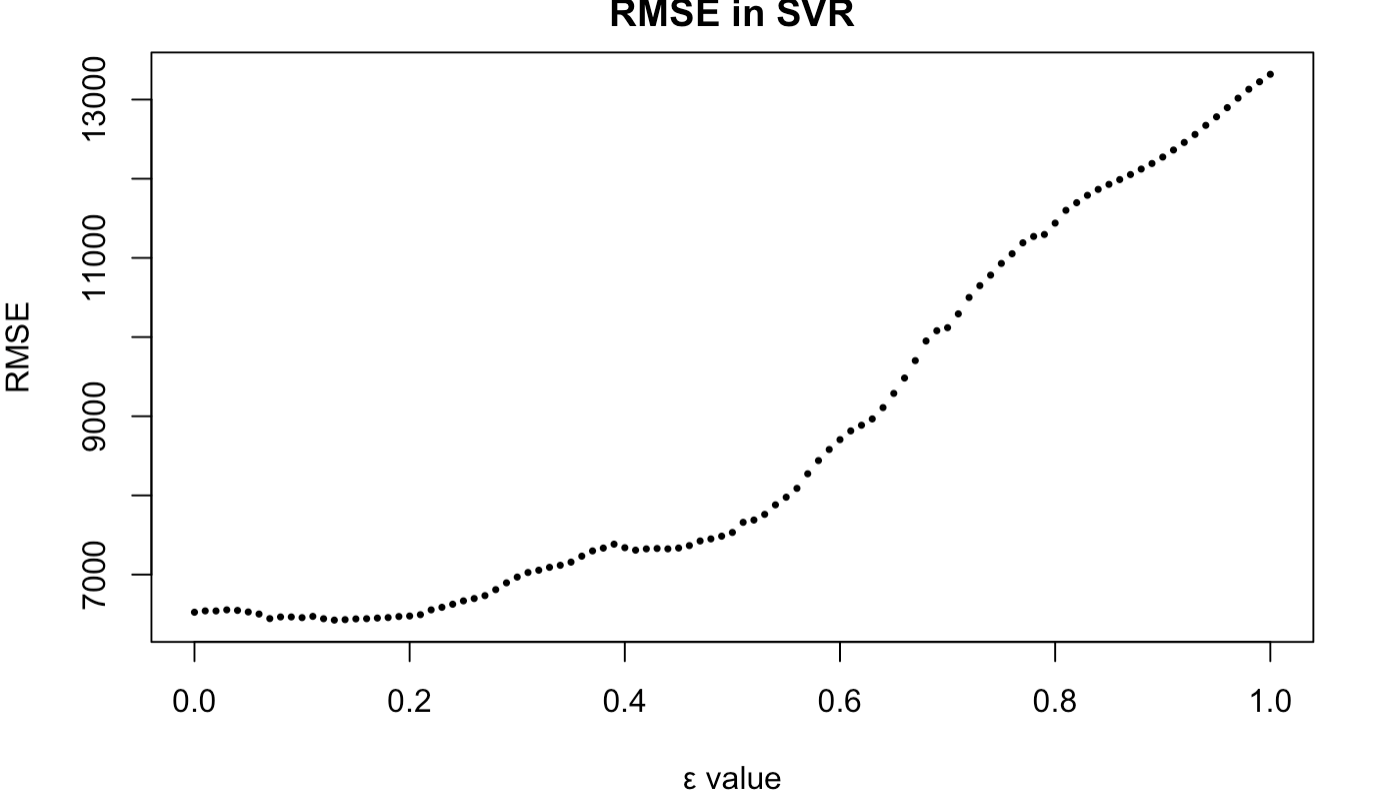
Fig3

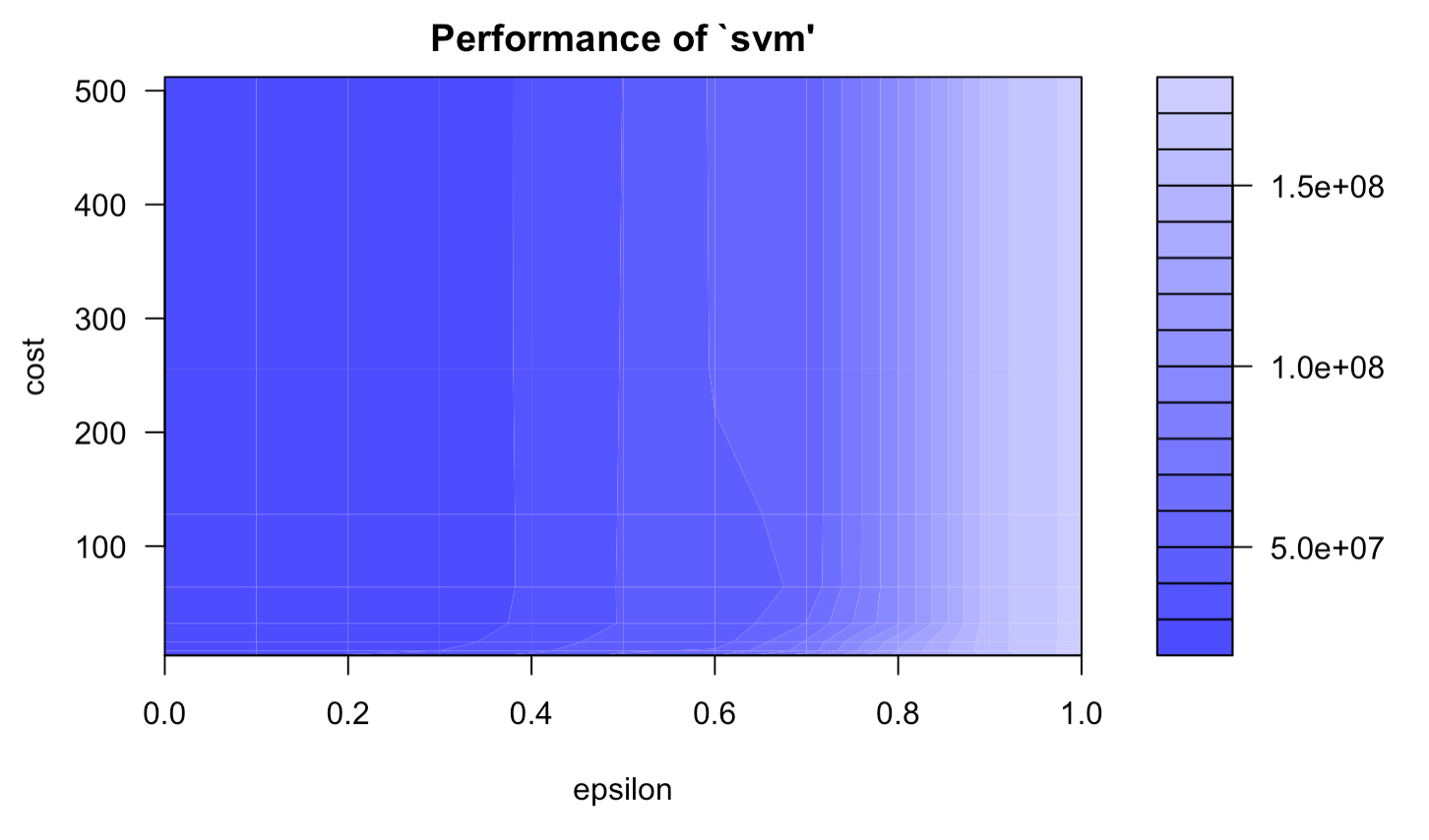
(red line : the actual data, blue line: the forecast result of next 14days, greyer zone : 80% CI, lighter Zone : 95%CI)

Fig4

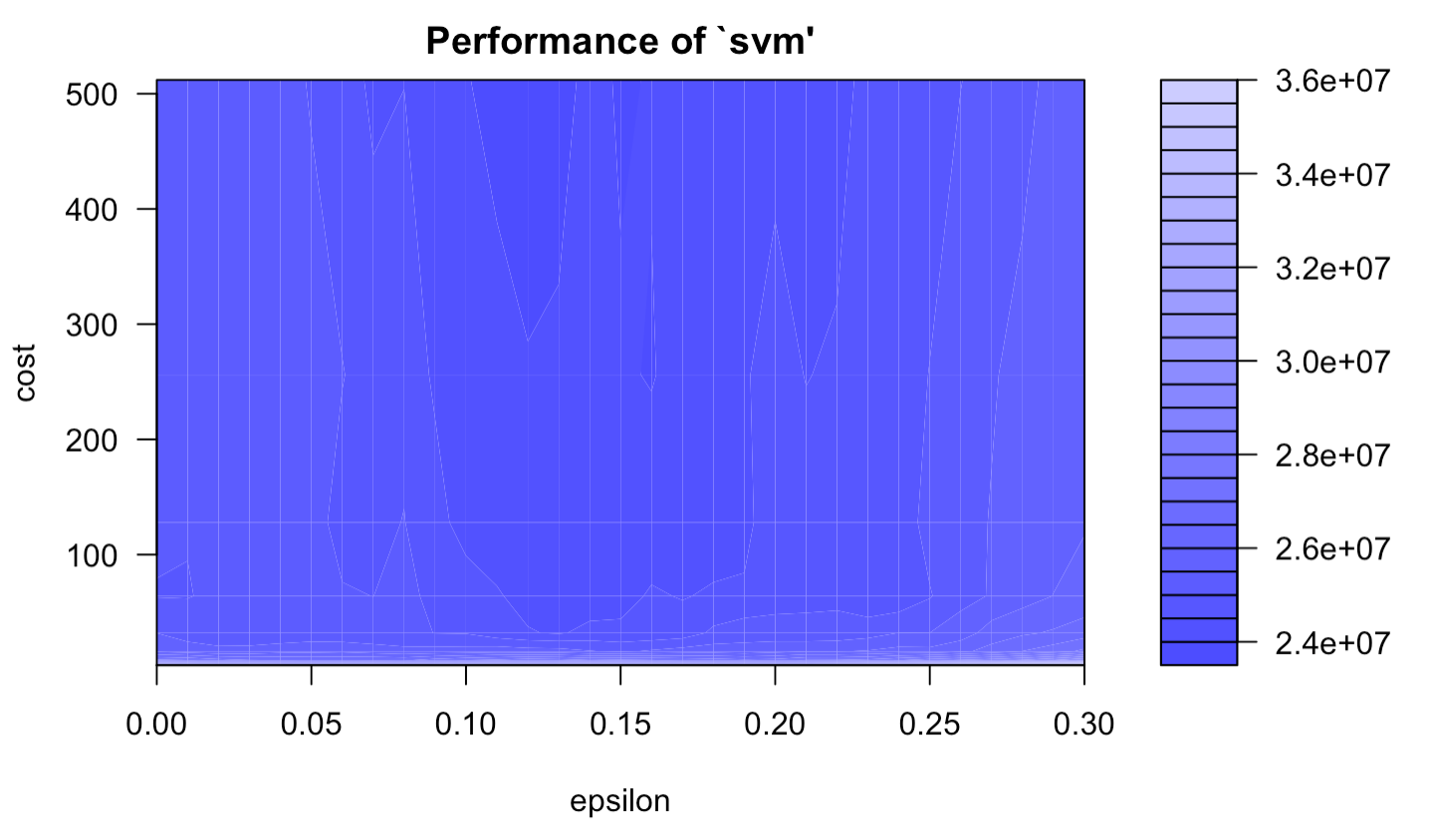
(black points= actual value , blue points = predict value)

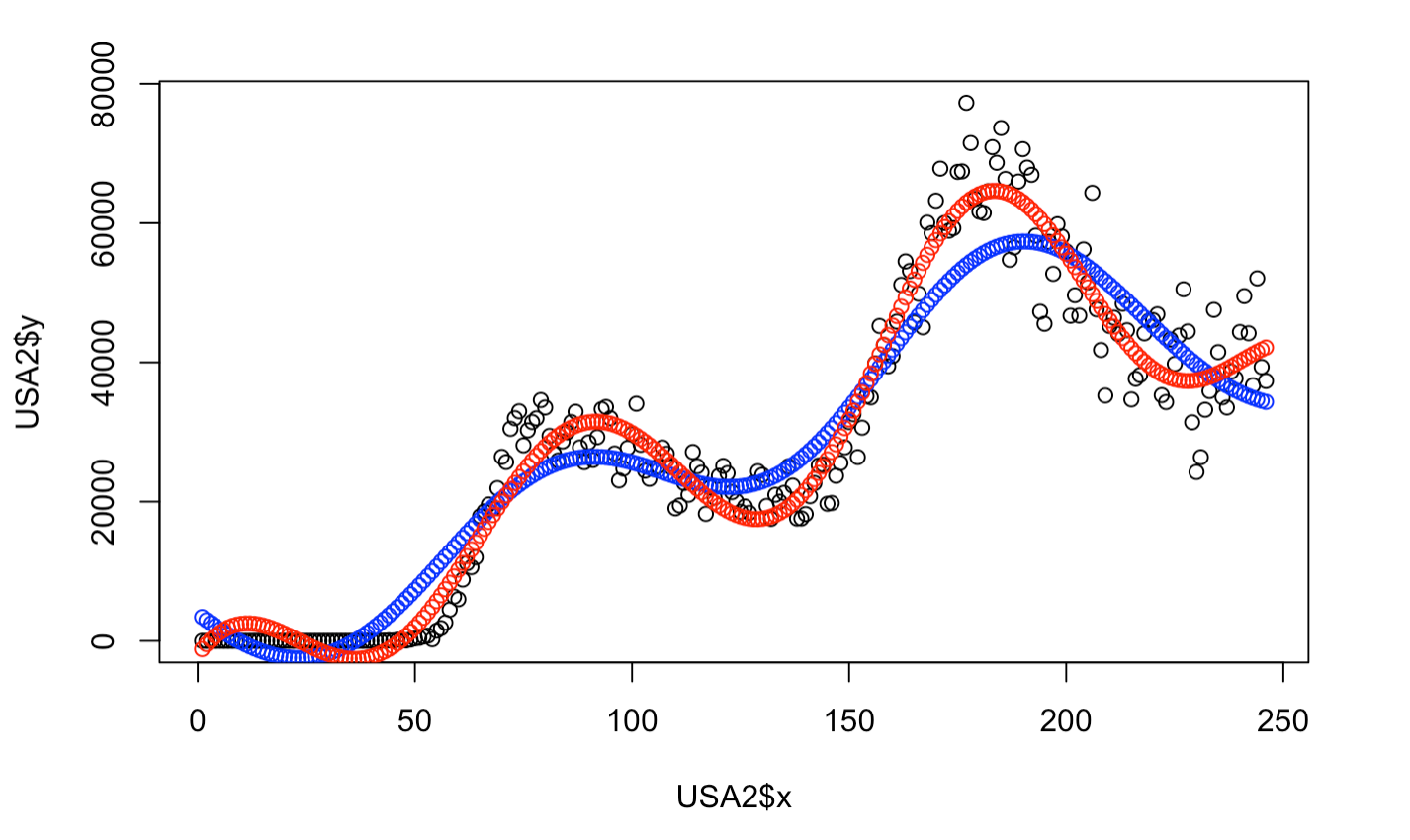
Fig5

Fig6

Fig7

On this graph we can see that the darker the region is the better our model is (because the RMSE is closer to zero in darker regions).

Fig8

Fig9

(black= actual value, blue = rough model, red= best model)

**R-codes**

covid<-read.csv("~/Desktop/archive/time\_series\_covid\_19\_confirmed\_US.csv")

US=covid[,-(1:11)]

USA=colSums(US)

USA=rbind(US,USA)

USA1=USA[3341,]

UStotal=USA[3341,]

for (i in 2:246) {

if(UStotal[1,i]>=0){

USA1[1,i]=UStotal[1,i]-UStotal[1,i-1]

}

}

USA1day=t(ts(USA1))

USA2=data.frame(x=1:246,y=USA1day)

train=window(USA1day,start=1,end=232)

test=window(USA1day,start=233)

#ARIMA

arimaAuto=auto.arima(train)

arimaAuto

forecastARIMA= forecast(arimaAuto,h=14)

plot(forecastARIMA)

lines(USA1day,col="red")

accuracy(forecastARIMA,test)

#Holt's exp smoothing

holtsForecast=HoltWinters(train,gamma = F)

holtsForecast

plot(holtsForecast)

forecasteplot=forecast:::forecast.HoltWinters(holtsForecast,h=14)

plot(forecasteplot)

lines(USA1day, col="red")

accuracy(forecasteplot,test)

#SVR

svr=svm(y~x,USA2)

svr.pred=predict(svr,USA2)

plot(USA2$x,USA2$y)

points(svr.pred,col="blue")

RMSE(USA2$y,svr.pred)

#SVr Tune

numSV=sapply(X=seq(0,1,0.01),FUN=function(e)svm(X3341~x, USA2, cost=1, epsilon =e)$tot.nSV)

plot(x=seq(0,1,0.01), y=numSV, xlab="ε value", ylab="# of support vectors", pch=16, cex=.5, main="# of SVs in SVR")

RMSE = sapply(X=seq(0,1,0.01),

FUN=function(e) sqrt(mean((svm(X3341~x, USA2, cost=1, epsilon =e)$residuals)^2)))

plot(x=seq(0,1,0.01), y=RMSE, xlab="ε value", ylab="RMSE", pch=16, cex=.5, main="RMSE in SVR")

tuneSVR=tune(method=svm,X3341~x,data=USA2,ranges =list(epsilon =seq(0,1,0.1), cost= 2^(2:9)))

print(tuneSVR)

plot(tuneSVR)

tuneSVR=tune(method=svm,X3341~x,data=USA2,ranges =list(epsilon =seq(0,0.3,0.01), cost= 2^(2:9)))

print(tuneSVR)

plot(tuneSVR)

tunedModel <- tuneSVR$best.model

tunedModelY <- predict(tunedModel, USA2)

rmse(USA2$X3341,tunedModelY)

plot(USA2$x,USA2$y)

points(svr.pred,col="blue")

points(tunedModelY,col="red")

RMSE(USA2$y,tunedModelY)