**Solution Sheet**

1. Which model have you used for Covid Cases prediction? Explain your model.

We have implemented different linear and non linear statistical machine learning methodologies to predict covid cases. Also we’ve used deep neural networks to fit the model. But Gradient boosting algorithm found out to be best.

First we have cleaned the training data and filled the empty values by using simple linear regression rather than removing whole rows. And removed some columns having larger number of missing values (i.e. >100). Then after removing outliers, one hot encoding categorical variables and normalization, we have analyzed the data. Determined the correlation of independent variables with target variable and performed ANOVA test on categorical variables

In model development we’ve implemented different machine learning algorithms to fit our models. Here are the results.

(We’ve split the training dataset into 2 parts training set and validation set in order to get validation R2 score)

Algorithm : Linear Regression

The R-square on training set is: 0.6432875378541507

The R-square on validation set is: -1.73826303242516e+22

Algorithm : Random Forest Regressor

The R-square on training set is: 0.9206984282727091

The R-square on validation set is: 0.5967139787265956

Algorithm : Gradient Boosting Algorithm

The R-square on training set is: 0.9953268744832343

The R-square on validation set is: 0.6160192968512226

Algorithm : SVM Regressor

The R-square on training set is: -0.34785909030958306

So we have finalized the model trained gradient boosting algorithm with hyper parameters :

n\_estimators: 625,

max\_depth: 7,

min\_samples\_split: 7,

learning\_rate: 0.01,

loss: ls

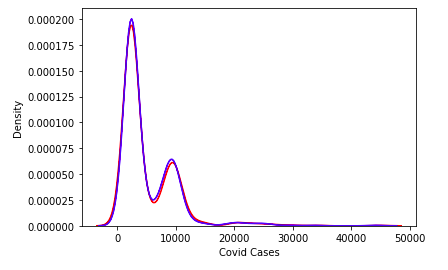
For deployment we’ve trained the model with same hyper parameters on whole dataset

The R-square on whole data is: **0.9929334313973398**

The mean square error of price and predicted value on whole

data is: **162463.8004209902**

Distribution plot:

****

1. Which model have you used for Foreign Visitors Time series prediction? Explain your model.

We’ve used stacked LSTM (Long Short Term memory) model to fit the data.

First we have cleaned the training data and filled the empty values by using simple linear regression rather than removing whole rows. After cleaning, we’ve converted the pandas series to stacked numpy array with each element containing four time step inputs.

We’ve developed model in Keras framework. The stacked LSTM model contains five LSTM layers with 1024, 512, 256, 128, 64 hidden LSTM cells receptively. The last layer output is passed through three fully connected dense layers to get a single value output.

The model converges on 174th epoch. The model with minimum validation loss is selected for deployment.

Training Loss: **193.76837158203125**

Validation Loss: **204.8126220703125**

Model architecture:

Model: "sequential\_8"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

lstm\_32 (LSTM) (None, 4, 1024) 4202496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_33 (LSTM) (None, 4, 512) 3147776

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_34 (LSTM) (None, 4, 256) 787456

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_35 (LSTM) (None, 4, 128) 197120

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_36 (LSTM) (None, 64) 49408

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_22 (Dense) (None, 32) 2080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_23 (Dense) (None, 16) 528

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_24 (Dense) (None, 1) 17

=================================================================

Total params: 8,386,881

Trainable params: 8,386,881

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

With this model we’ve predicted the number of Foreign Visitors on 1st October 2020. Then we’ve predicted the covid cases on 1st October with already trained regression model.