Tanmay Gupta
Tg289

Covid-19 Data Analysis

Description-

Covid-19 is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first case was identified in December 2019. It has since spread worldwide, leading to an ongoing pandemic. The purpose of this project is to analyze the Covid-19 dataset from here- https://covidtracking.com/data/national and identify which algorithm works best to predict the number of COVID-19 daily cases in the United States of America. We will also see the potential problems in the dataset and the ways of correcting them.

Background-

The problem persists from December 2019 when first case was recorded, WHO gave an emergency in January 2020 that it is a contagious disease and with the widespread increase in the whole world, it was declared a pandemic by WHO is March 2020. The symptoms of the disease are fever, cough, fatigue, loss of smell and taste. The symptoms occur 1-14 days after contact with the virus. Preventive measures- Social Distancing, Wearing Masks, Sanitizer, Face Coverings.
Our main concern is the effect of the disease in the United States of America. The first Covid-19 positive case was recorded on January 20,2020 in Washington and a total of 15 Million cases have been recorded till date. With highest being on December 3rd, 2020 being 219,187. The most affected states are California, Texas and Florida with a combined number of cases of 4 Million. In this report we will examine the effect of our predictor variables which are Total people on ventilator, number of people hospitalized, negative cases, total negative cases, negative cases in a day on our response variable total number of positive cases in a day. We will use several algorithms and see the one which works best on our dataset.

Potential Solution-

The first thing is to understand the problems of the dataset and correct them. There are two main problems with our dataset.

1) As the number of Covid-19 positive and negative cases have increased over time, so the dataset is imbalanced, and we need to correct this imbalance, to get better predictions on our test set. We will see how to improve it before building the model.
2) The differences between the variables is very high. For example- The total number of positive cases is 14,534,035, and the negative is 161,986,294 which is more than 10 times the positive cases. So, we need to scale the data for the model to perform better.

With the balanced and scaled dataset, we can create static models to validate the correlation in the number of cases in a day with other variables and to see which models perform best.

Exploring the Dataset

The dataset has a total of 18 variables including our response variable. As the data is for the whole year from January 21,2020, so there are a total of 320 observations or rows, of which we split into the training and testing set. We use a 80:20 split, so there are 256 training observations and 64 testing observations.

Our Response or target Variable-

Today_positive_cases- The number of Covid-19 positive cases in a day.

Our drivers or predictor variables-

Date- The date on which the data was collected.
Total_Death- The total number of COVID-19 deaths on the recorded date.
Today_Death- The number of COVID-19 deaths in a day.
In_ICU_cummalative- Total Cumulative patients in ICU.
Total_in_ICU- The total number of people in the ICU.
Today_hospitalized- The number of people hospitalized on that day.
Total_hospitalized- The total number of people hospitalized.
Hospitalized_cummalative- Total cumulative COVID-19 hospitalized patients.
Total_negative- The total number of COVID-19 negative cases.
Today_negative- The number of negative COVID-19 cases on that day.
Ventilator_cummalative- Total cumulative number of people on the ventilator.
Total_on_ventilator- The total number of people on ventilator.
Total_positive_cases- The total number of positive COVID-19 cases.
Today_positive_cases- The number of positive COVID-19 cases on that day.
Total_people_recovered- The total number of people recovered from COVID-19.
Total_tests_results- The total test happened in the country.
Total_tests_results_today- The total tests done in a single day.

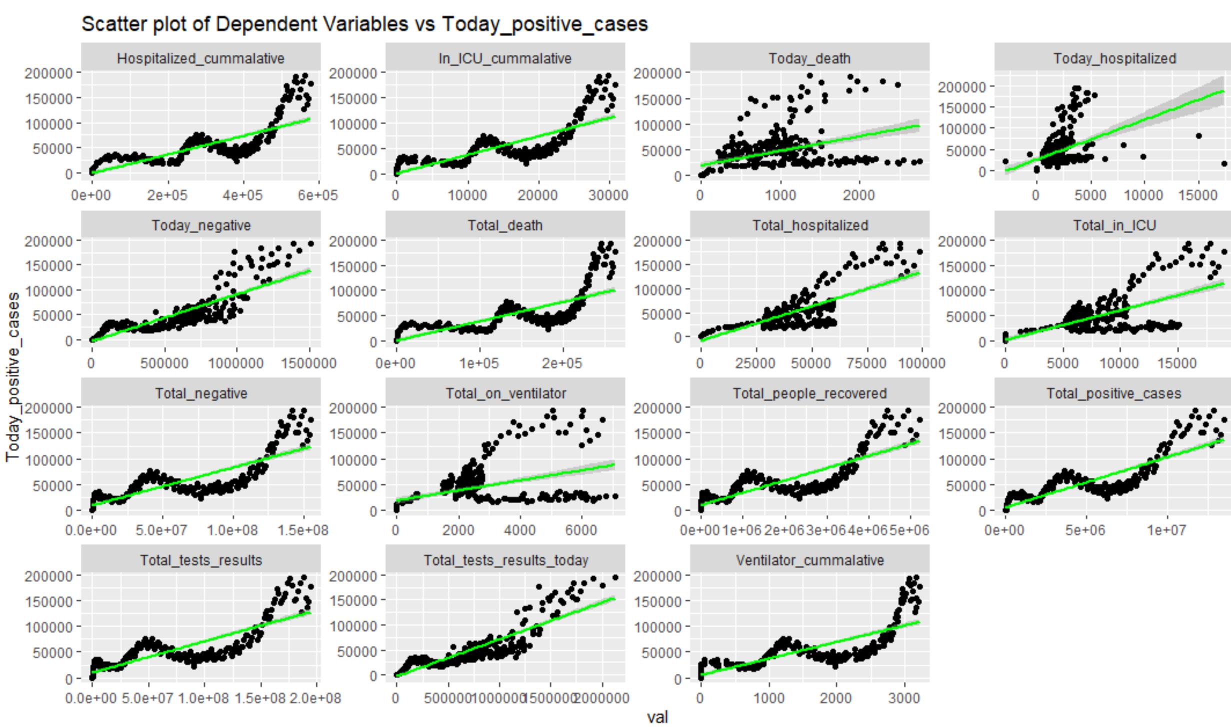
We will be using the predictor variables to predict our response variable. We will see the significance of the variables in the linear regression model.

Let’s take a **glimpse** at the dataset-

```
Date                      <fct> 2020-12-06, 2020-12-05, 2020-12-04, 2020-12-03, 2020-12-02, 2020-12-01, 2020-11-30, 2020-11-29, 2020-11-28, 2020-11-27, 2020-11-26, 2020-11-25, 2020-11-24, 2020-11-23, 2020-11-22, 2020-11-...
Total_death               <int> 273374, 272236, 269791, 267228, 264522, 261789, 259316, 258180, 257377, 256132, 254760, 253424, 251135, 249069, 248114, 247216, 245704, 243820, 241853, 239989, 238434, 237815, 237128, 2357...
Today_death               <int> 1138, 2445, 2563, 2706, 2733, 2473, 1136, 803, 1245, 1372, 1336, 2289, 2066, 955, 898, 1512, 1884, 1967, 1864, 1555, 619, 687, 1331, 1268, 1119, 1563, 1360, 577, 496, 1119, 1173, 1146, 111...
In_ICU_cummalative        <int> 31946, 31831, 31608, 31276, 31038, 30749, 30469, 30274, 30109, 29858, 29673, 29540, 29245, 28990, 28828, 28693, 28472, 28216, 27989, 27681, 27437, 27269, 27172, 26997, 26803, 26584, 26335,...
Total_in_ICU              <int> 20145, 19950, 19858, 19723, 19680, 19295, 18801, 18437, 18249, 18061, 18019, 17740, 17314, 17080, 16411, 16264, 16138, 15763, 15558, 15018, 14494, 13849, 13491, 13279, 12911, 12625, 12071,...
Today_hospitalized        <int> 2256, 3316, 4652, 5331, 5028, 5222, 3394, 2429, 3404, 3499, 2247, 4568, 4591, 2985, 2291, 3340, 3891, 4420, 4384, 3812, 3033, 1868, 3424, 3686, 3551, 3275, 4033, 2261, 1467, 2297, 3143, 25...
Total_hospitalized        <int> 101487, 101190, 101276, 100755, 100322, 98777, 96149, 93357, 91762, 89950, 90564, 90041, 88174, 85979, 83882, 83346, 82279, 80682, 79517, 77079, 73377, 70202, 69588, 68599, 67233, 65557, 6...
Hospitalized_cummalative  <int> 585676, 583420, 580104, 575452, 570121, 565093, 559871, 556477, 554048, 550644, 547145, 544898, 540330, 535739, 532754, 530463, 527123, 523232, 518812, 514428, 510616, 507583, 505715, 5022...
Total_negative            <int> 161986294, 160813704, 159286709, 158026052, 156787587, 155805555, 153863841, 152644033, 151760885, 150483950, 148979519, 148129101, 146968914, 145615236, 144267577, 143013286, 141583409, 1...
Today_negative            <int> 1172590, 1526995, 1260657, 1238465, 982032, 1941714, 1219808, 883148, 1276935, 1504431, 850418, 1160187, 1353678, 1347659, 1254291, 1429877, 1389889, 1286682, 1074535, 1092466, 1103588, 11...
Ventilator_cummalative    <int> 3322, 3321, 3305, 3280, 3252, 3223, 3205, 3184, 3179, 3171, 3153, 3147, 3123, 3106, 3094, 3087, 3078, 3052, 3024, 3004, 2988, 2975, 2972, 2967, 2954, 2947, 2922, 2907, 2900, 2898, 2885, 28...
Total_on_ventilator       <int> 7094, 7005, 6999, 6867, 6855, 6649, 6520, 6245, 6148, 6030, 5986, 5990, 5630, 5458, 5233, 5103, 5055, 4859, 4701, 4379, 4166, 3939, 3945, 3769, 3622, 3367, 3201, 3110, 2977, 2947, 2850, 28...
Total_positive_cases      <int> 14534035, 14357264, 14146191, 13921360, 13711156, 13515360, 13338607, 13191020, 13055778, 12901256, 12707531, 12581196, 12397567, 12230553, 12079595, 11928754, 11749039, 11556239, 11373404...
Today_positive_cases      <int> 176771, 211073, 224831, 210204, 195796, 176753, 147587, 135242, 154522, 193725, 126335, 183629, 167014, 150958, 150841, 179715, 192800, 182835, 163972, 156732, 148873, 144432, 163538, 1721...
Total_people_recovered    <int> 5624444, 5576026, 5470389, 5404018, 5322128, 5226581, 5146643, 5065148, 5024447, 4947446, 4871312, 4835956, 4696996, 4633600, 4526635, 4529788, 4457930, 4410709, 4348089, 4292094, 4242683,...
states                    <int> 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, ...
Total_tests_results       <int> 204063869, 202429337, 200259581, 198404712, 196576482, 195117280, 192776284, 191173031, 189883061, 188173495, 186047207, 184877012, 183156307, 181366035, 179607475, 177862987, 175858750, 1...
Total_tests_results_today <int> 1634532, 2169756, 1854869, 1828230, 1459202, 2340996, 1603253, 1289970, 1709566, 2126288, 1170195, 1720705, 1790272, 1758560, 1744488, 2004237, 1919404, 1821019, 1539051, 1511767, 1486671,...
```

Here we can see the first few rows of each column in our data frame. It also tells us about the class of the feature. As the data is related to number of cases, so all variables are int except the Date variable.

Now let’s look at the **distribution** of the dataset-



We have plotted our response variable with all the predictor variables and the green line here shows the linear fitting. We can see a non-linear association of most of the variables. Today_hospitalized is inclined to one part of the graph. And the distribution of Today_death and Total_on_ventilator is weird, and it doesn’t tell anything about the distribution.

Correlation between variables-

	Total_death	Today_death	In_ICU_cummalative	Total_in_ICU	Today_hospitalized	Total_hospitalized	Hospitalized_cummalative	Total_negative	Today_negative
Total_death	1.00	0.29	0.99	0.58	0.27	0.67	1.00	0.94	0.96
Today_death	0.29	1.00	0.26	0.83	0.59	0.75	0.32	0.21	0.30
In_ICU_cummalative	0.99	0.26	1.00	0.56	0.26	0.65	0.99	0.98	0.96
Total_in_ICU	0.58	0.83	0.56	1.00	0.55	0.96	0.60	0.50	0.58
Today_hospitalized	0.27	0.59	0.26	0.55	1.00	0.54	0.29	0.24	0.30
Total_hospitalized	0.67	0.75	0.65	0.96	0.54	1.00	0.69	0.59	0.70
Hospitalized_cummalative	1.00	0.32	0.99	0.60	0.29	0.69	1.00	0.95	0.96
Total_negative	0.94	0.21	0.98	0.50	0.24	0.59	0.95	1.00	0.92
Today_negative	0.96	0.30	0.96	0.58	0.30	0.70	0.96	0.92	1.00
Ventilator_cummalative	0.98	0.24	1.00	0.53	0.25	0.61	0.98	0.99	0.95
Total_on_ventilator	0.33	0.81	0.29	0.91	0.48	0.83	0.35	0.21	0.33
Total_positive_cases	0.95	0.26	0.98	0.56	0.27	0.64	0.96	1.00	0.93
Today_positive_cases	0.79	0.43	0.83	0.69	0.43	0.80	0.81	0.84	0.86
Total_people_recovered	0.94	0.21	0.97	0.51	0.24	0.58	0.94	1.00	0.91
states	0.56	0.55	0.50	0.62	0.41	0.64	0.56	0.40	0.55
Total_tests_results	0.94	0.21	0.97	0.51	0.25	0.59	0.95	1.00	0.91
Total_tests_results_today	0.96	0.31	0.97	0.59	0.31	0.70	0.96	0.95	0.99

	Ventilator_cummalative	Total_on_ventilator	Total_positive_cases	Today_positive_cases	Total_people_recovered	states	Total_tests_results	Total_tests_results_today
Total_death	0.98	0.33	0.95	0.79	0.94	0.56	0.94	0.96
Today_death	0.24	0.81	0.26	0.43	0.21	0.55	0.21	0.31
In_ICU_cummalative	1.00	0.29	0.98	0.83	0.97	0.50	0.97	0.97
Total_in_ICU	0.53	0.91	0.56	0.69	0.51	0.62	0.51	0.59
Today_hospitalized	0.25	0.48	0.27	0.43	0.24	0.41	0.25	0.31
Total_hospitalized	0.61	0.83	0.64	0.80	0.58	0.64	0.59	0.70
Hospitalized_cummalative	0.98	0.35	0.96	0.81	0.94	0.56	0.95	0.96
Total_negative	0.99	0.21	1.00	0.84	1.00	0.40	1.00	0.95
Today_negative	0.95	0.33	0.93	0.86	0.91	0.55	0.91	0.99
Ventilator_cummalative	1.00	0.25	0.99	0.82	0.98	0.47	0.98	0.96
Total_on_ventilator	0.25	1.00	0.28	0.45	0.22	0.54	0.22	0.33
Total_positive_cases	0.99	0.28	1.00	0.87	1.00	0.43	1.00	0.96
Today_positive_cases	0.82	0.45	0.87	1.00	0.85	0.43	0.85	0.89
Total_people_recovered	0.98	0.22	1.00	0.85	1.00	0.39	1.00	0.94
states	0.47	0.54	0.43	0.43	0.39	1.00	0.39	0.51
Total_tests_results	0.98	0.22	1.00	0.85	1.00	0.39	1.00	0.94
Total_tests_results_today	0.96	0.33	0.96	0.89	0.94	0.51	0.94	1.00

Here we can see the correlation between the variables. The values are between 0 and 1. If the value is more than 0.5 it shows a strong correlation between those variables.

For example-

- Total_test_results has a positive and strong correlation with the number of positive COVID-19 cases in a day. So, as the number of total test results increases, so we can see an increase in the per day COVID-19 cases.

- Today_negative also has a strong positive correlation with today_positve_cases. As the number of negative cases increase, so does the positive cases increase.

- Today_deaths have weak positive correlation with today_positive_cases, so the increase in per day deaths doesn't affect much on the per day positive cases.

- We can also see a weak positive correlation between the total number od deaths in a day and the total results in a day. The effect won't be much on either of them.

Addressing the **limitations** or **problem** with the data

For imbalance-

- We used the smoteR for correcting the imbalance in the dataset. SmoteR (Torgo et al., 2013) is an adaption for regression of the well-known Smote (Chawla et al., 2002) algorithm. Earlier these techniques existed only for classification problem where there is imbalance between two classes and one class outnumber the other. Mainly 3 techniques were introduced-
  - Under sampling- In this technique the non-rare case is under sampled, which means that the cases which are more in number in the variable is reduced.
  - Over sampling- In this technique the rare class in over sampled to make its effect equivalent to the non-rare class, which means that the cases which are less in number are increased.
  - SmoteR- SmoteR is the Smote for regression. In this technique, both under sampling and over sampling are performed. The rare cases are over sampled, and the non-rare case is over sampled, so as to keep a balance of both and the model doesn't favor any case.
- SmoteR oversamples the rare case in the data (which is defined by us in the function) and under samples the non- rare case. The weightage of under and over sampling is also to be provided to the function by us.
- We used SmoteRegress () function from the UBL library available in R.

For in-variable high difference-

- We scaled the dataset using scale () function available in R. scale compresses the dataset which is helpful when the order of magnitude in the variables is different. (like in our case).
- It calculates the mean and standard deviation of the vector, then "scale" each element by those values by subtracting the mean and dividing by the standard deviation.
- This scales the data so that there is not much difference and we can be sure that the variable with big value is not the one which is driving the model.

Model fitting and evaluation-

We fit various models on our training set and using the model we predict our test set. We then compare the model performance based on the MSE and accuracy (as it is a regression problem). Let's dive into our models.

1. Linear Regression Model with best subset selection.

We fit a linear regression model to our training data using the lm() function in R. Then we used the regsubsets() function to find the best subset on the 15 explanatory variables (excluding date and state). To select the best variables, we went with the stepwise selection using "forward selection", "backward selection" and a combination of both.

We select the model with the best AIC value (least AIC). Here you can see the model with best AIC value.

```
Step:  AIC=-1050.49
Today_positive_cases ~ Total_positive_cases + Total_hospitalized +
    Total_death + Total_tests_results_today + In_ICU_cummalative +
    Today_negative + Ventilator_cummalative + Total_on_ventilator +
    Today_death + Total_tests_results

                            Df Sum of Sq    RSS      AIC
<none>                                   3.3543 -1050.49
+ Total_negative             1   0.02122 3.3331 -1050.07
+ Hospitalized_cummalative   1   0.00212 3.3522 -1048.65
+ Total_in_ICU               1   0.00061 3.3537 -1048.54
+ Total_people_recovered     1   0.00007 3.3542 -1048.50
+ Today_hospitalized         1   0.00001 3.3543 -1048.50
- Total_tests_results        1   0.16092 3.5152 -1040.83
- Today_death                1   0.62193 3.9762 -1010.14
- Total_positive_cases       1   0.64645 4.0008 -1008.61
- Total_on_ventilator        1   0.72989 4.0842 -1003.47
- Ventilator_cummalative     1   0.81249 4.1668  -998.49
- Total_death                1   0.88461 4.2389  -994.21
- Today_negative             1   1.51447 4.8688  -959.72
- Total_hospitalized         1   2.21766 5.5720  -926.13
- Total_tests_results_today  1   2.63979 5.9941  -907.94
- In_ICU_cummalative         1   2.72721 6.0815  -904.34
```

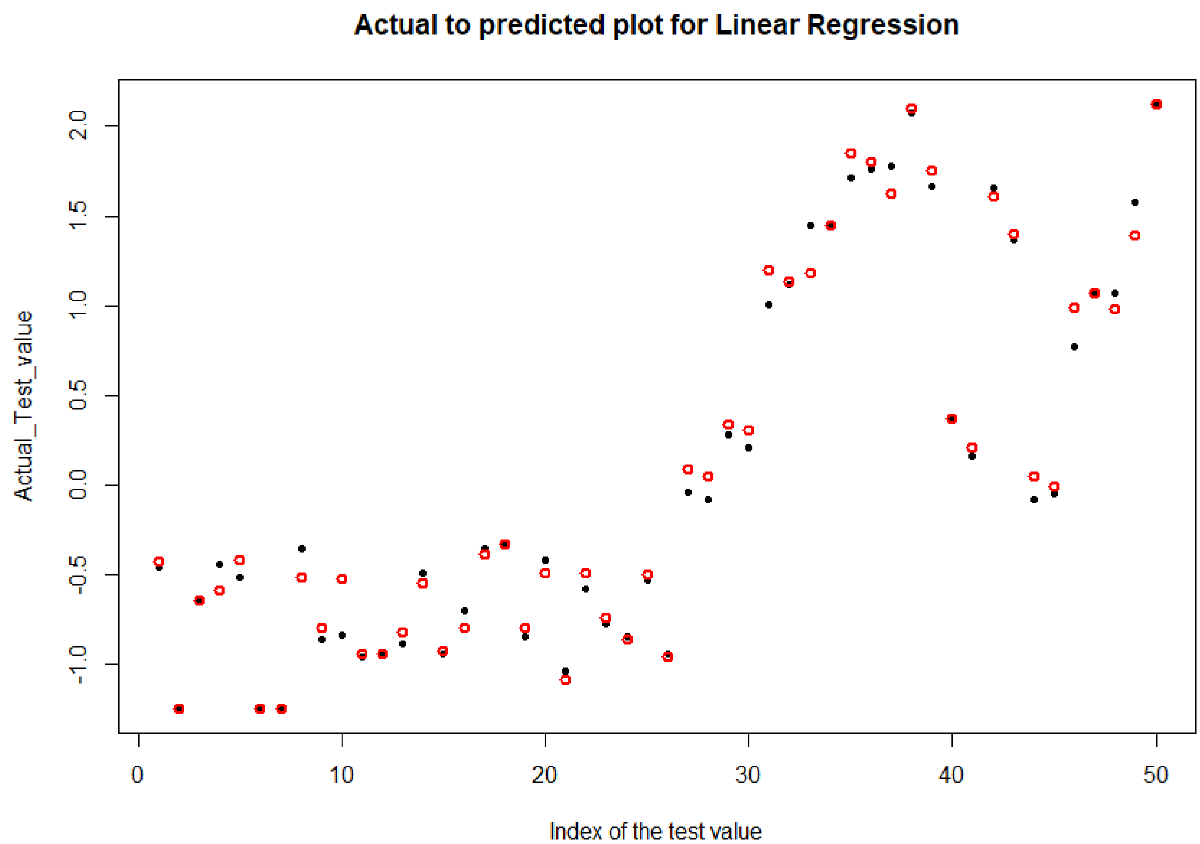
```
Call:
lm(formula = Today_positive_cases ~ Total_positive_cases + Total_hospitalized +
    Total_death + Total_tests_results_today + Today_negative +
    In_ICU_cummalative + Ventilator_cummalative + Total_on_ventilator +
    Today_death + Total_tests_results + Hospitalized_cummalative +
    Total_negative, data = scaled_data_1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.33071 -0.06844 -0.00005  0.05773  0.34053

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.194e-15  7.352e-03   0.000  1.00000
Total_positive_cases     -1.192e+00  2.939e-01  -4.056 6.78e-05 ***
Total_hospitalized        5.254e-01  4.352e-02  12.073  < 2e-16 ***
Total_death              -1.042e+00  3.613e-01  -2.884  0.00428 **
Total_tests_results_today  1.508e+00  1.087e-01  13.869  < 2e-16 ***
Today_negative           -9.639e-01  8.835e-02 -10.910  < 2e-16 ***
In_ICU_cummalative        3.918e+00  4.179e-01   9.375  < 2e-16 ***
Ventilator_cummalative   -1.504e+00  5.629e-01  -2.672  0.00806 **
Total_on_ventilator      -2.584e-01  4.660e-02  -5.545 7.85e-08 ***
Today_death               1.119e-01  1.543e-02   7.250 5.92e-12 ***
Total_tests_results       4.000e+00  1.472e+00   2.717  0.00707 **
Hospitalized_cummalative -7.254e-01  3.544e-01  -2.047  0.04179 *
Total_negative           -3.445e+00  1.850e+00  -1.862  0.06385 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

After we evaluated the stepwise selection model, we use the significant variables to create the model. We use this new model to predict values of our test set.
We record the performance of the model by using the MSE and the accuracy.

For **linear regression model** we get an **MSE of 0.1037821** and a **correlation accuracy of 0.926** which is good for a correlated dataset as ours. The lower the MSE the better the model and the higher the correlation accuracy the better the model.

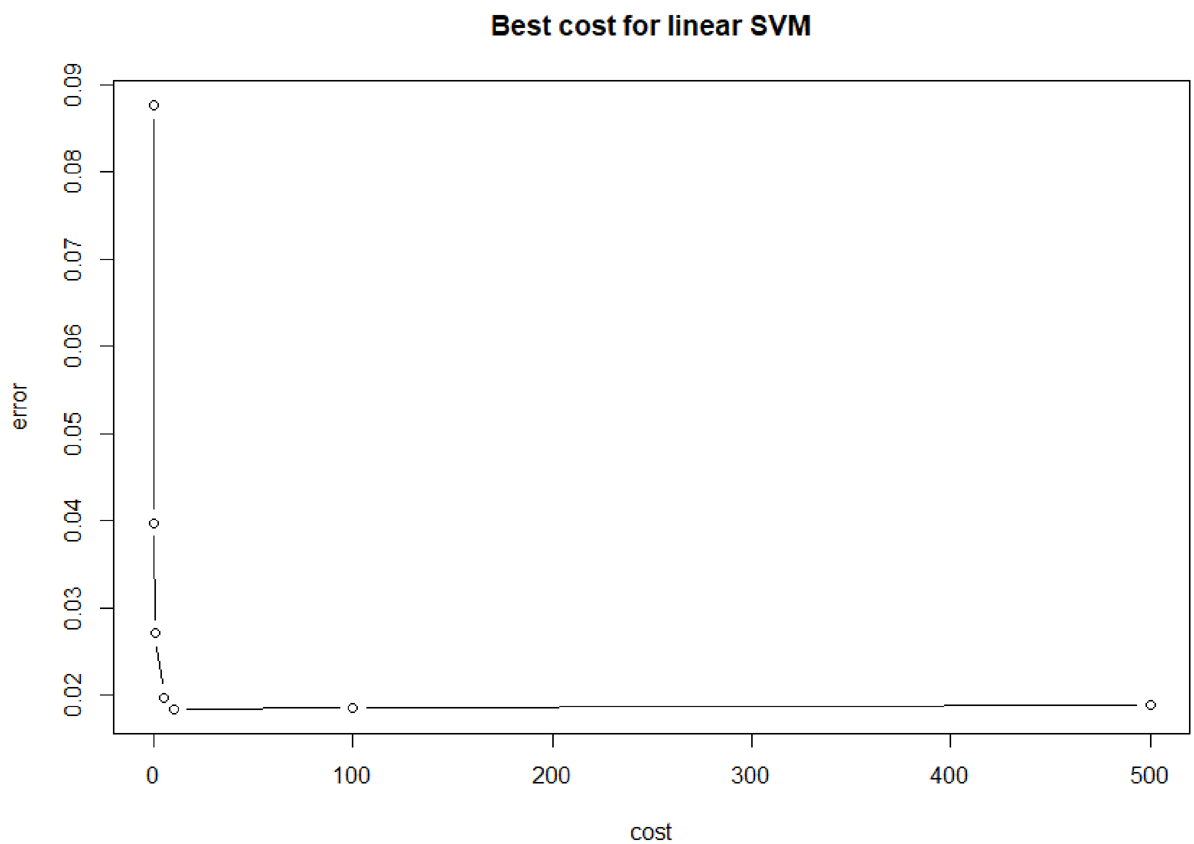


The plot shows the actual and the predicted points for the model. The black dots are the actual points and the red ones are the predicted points from our model. We can see that out of 65 test cases 4-5 were correctly predicted and the rest are near.

2. Linear Kernel Support Vector Machine

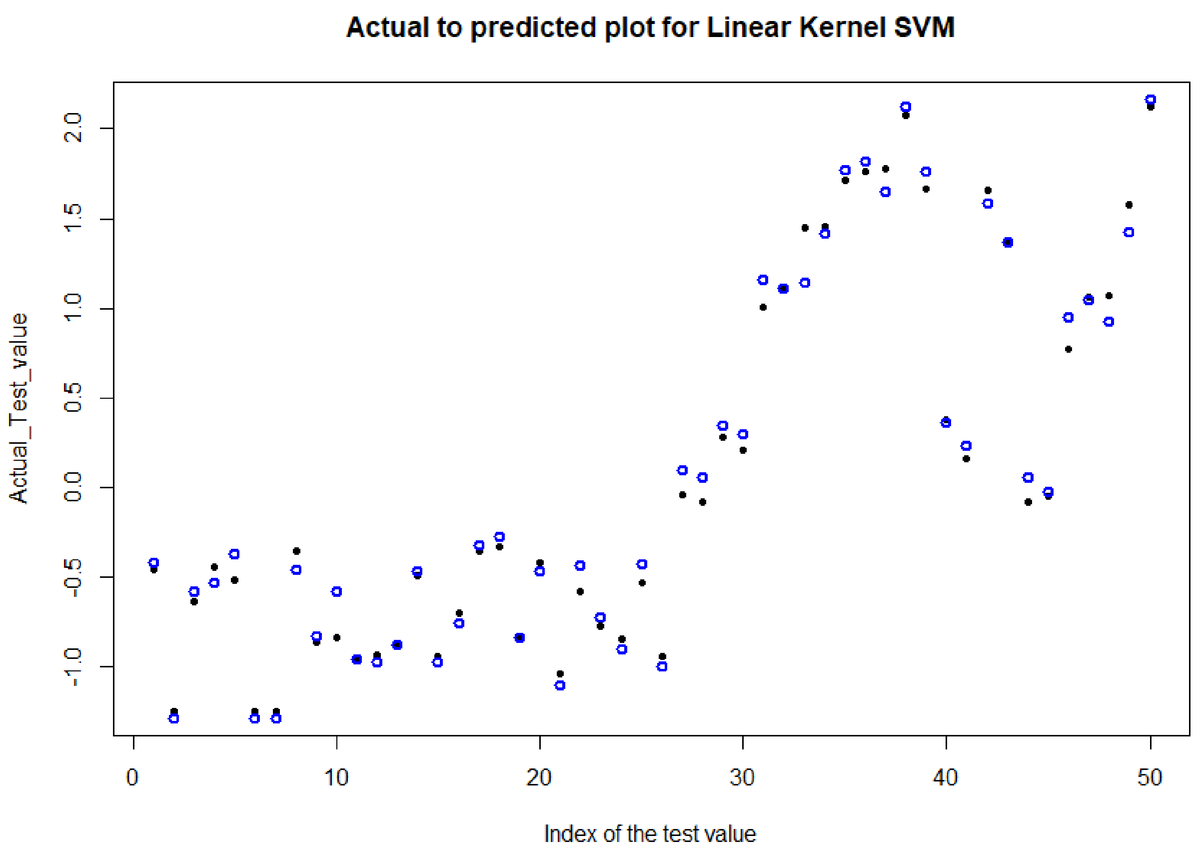
We fit a linear SVM model to our training data with the defined parameters (C=1, sigma(gamma) = 0.067 and epsilon= 0.1) and number of support vectors=104. Now we tune our model to get the best parameters, we used the “tune” function from the “e1071” library (library for SVM in R). After tuning the model, we got the best parameters which were “C= 10”.

The plot below shows the Best cost for linear SVM with ‘error’ on Y axis and the value of ‘Cost’ (c) on X axis. The error is minimum at C= 10



Now we fit another model with the tuned parameters and used this new model to predict the values on our test set. The results are somewhat better than the linear regression model because the SVM fits a margin around the hyperplane and then predict the values.

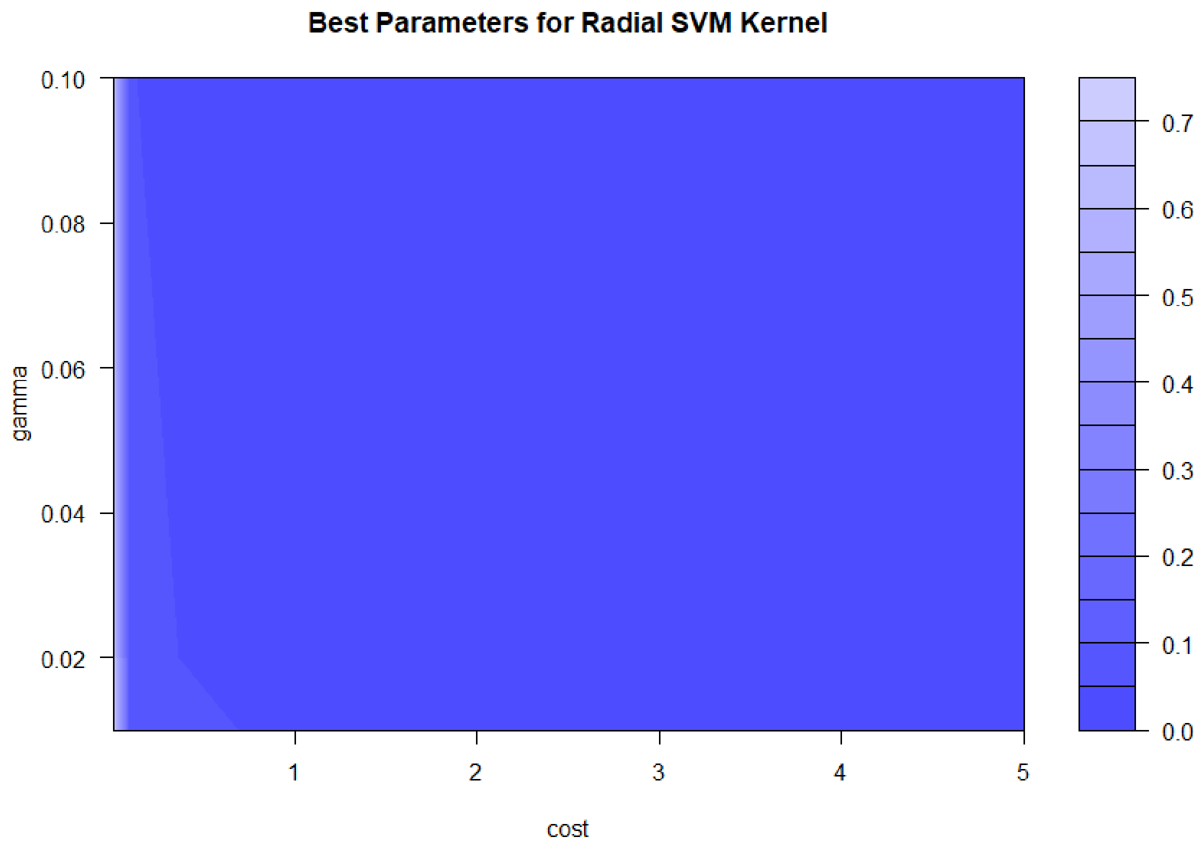
For the Linear Kernel SVM model we got an **MSE of 0.097984** and **correlation accuracy of 0.934**. The MSE of Linear SVM is lower than Linear Regression and the accuracy is higher than Linear Regression.



The plot shows the actual and the predicted points for the model. The black dots are the actual points and the blue ones are the predicted points from our model. We can see that the predicted points are close to actual ones. Hence it has performed better than Linear Regression.

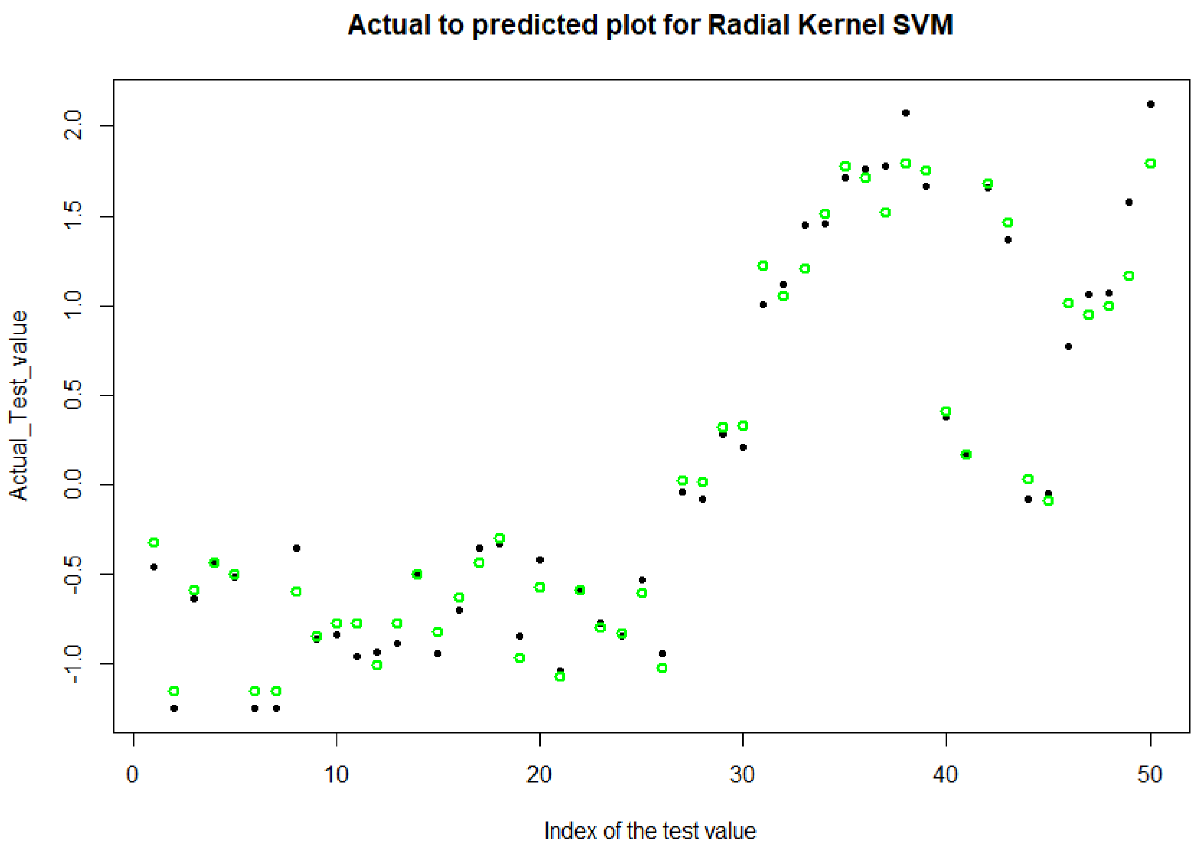
3. Radial Kernel Support Vector Machine

We fit the Radial Kernel Support Vector Machine (SVM) to our training set with the defined parameters (C=1, sigma(gamma) = 0.07 and epsilon= 0.1). Now we tune our model to get the best parameters, we use the "tune" function from the "e1071" library. After tuning the model, we got the best parameters which were "C= 5" and "gamma= 0.02", here's a plot of our tuning.



Here we can see that the least error goes at cost=5 and gamma value=0.02.

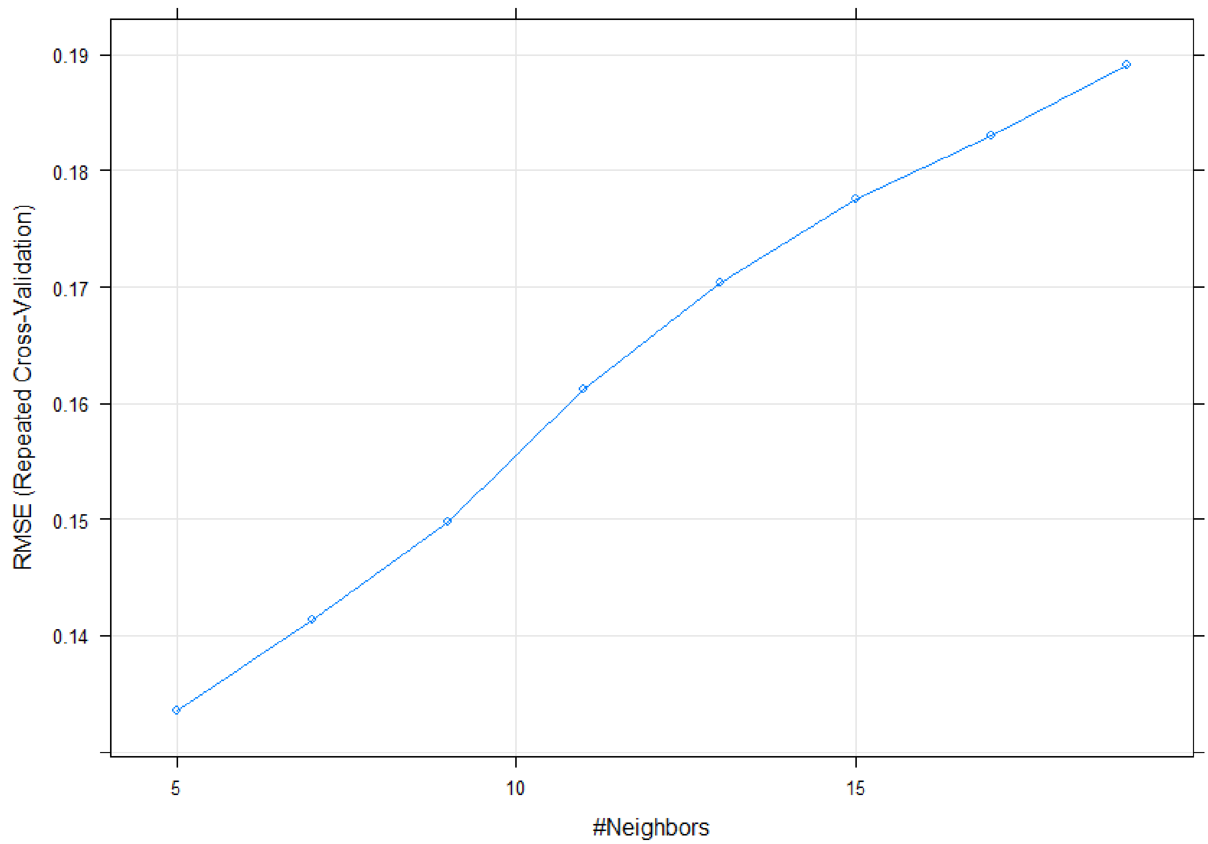
Now we use these parameters to train our model, then we make a prediction on our test values. We report an MSE of 0.1843159 and correlation accuracy of 0.873. This model doesn't perform as the other two linear models. Which means that our distribution or relationship between variables is changed by the SmoteR and scaling of the dataset. Hence a Linear model performs better on it.



Here's the performance plot of the predicted values of our Radial Kernel SVM model. The black dots are the actual values and the green ones are the predicted values. We can see many mismatched values which says about the performance of the model.

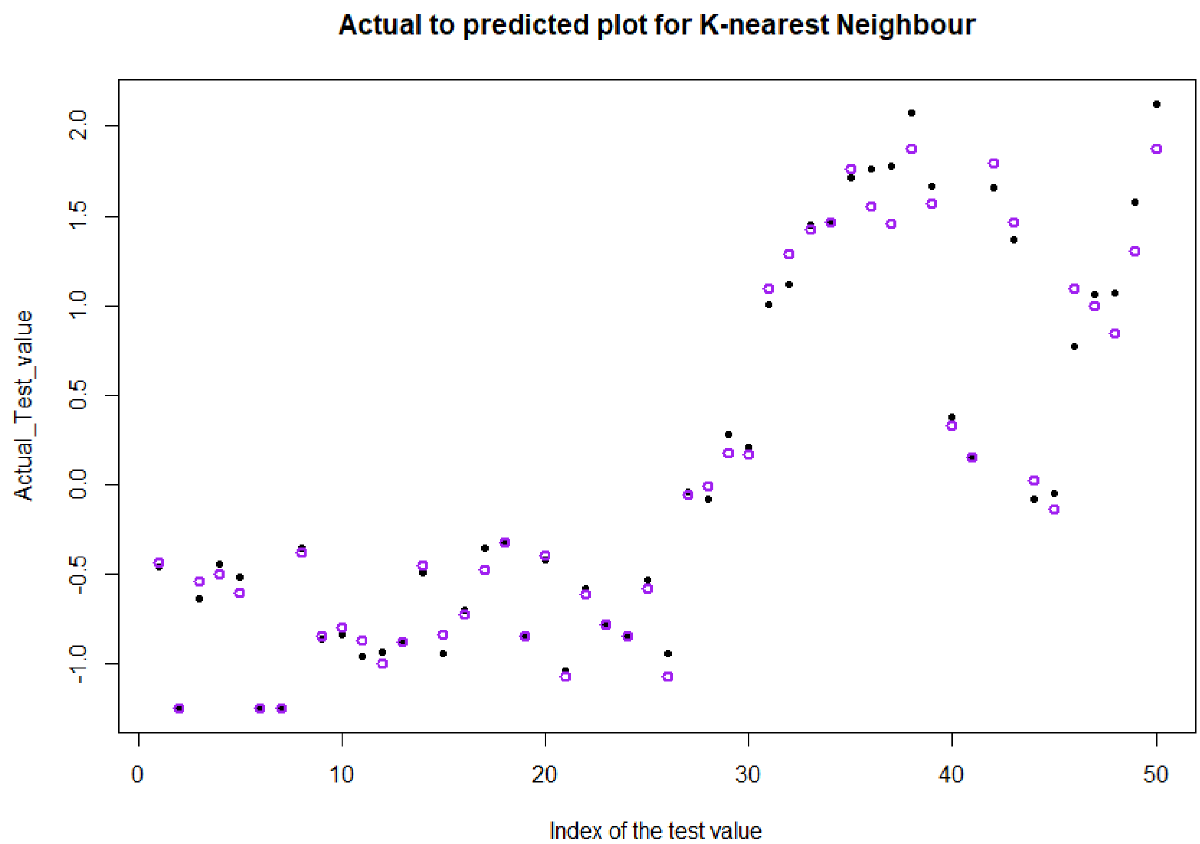
4. K-Nearest Neighbors (KNN)

For getting the best K-values, we first used 10-fold repeated cross-validation (repeated 3 times). So, we get 10 values of k and we select the one which gives the least error. Here for our model the best k value was k=5 which gives us the least RMSE.



Here in the plot we can see that we get the least error at K=5 and the error increases as the number of K increases.

Now we fit a model with K=5 and predict the values of test set using this model. The model performed well as compared to radial SVM but couldn’t outperform Linear Kernel SVM and Linear Regression. We recorded an MSE of .1349 and correlation accuracy of 0.902



This is the performance plot of KNN. Black are the actual and purple ones are the predicted values. We can see a fair prediction. We will see in the model comparison and conclusion which performance was best with least error.

Model Comparison

We will compare the models based on the MSE (mean squared error) and correlation accuracy. Here's a comparison table which gives a clear understanding of the performances.

Model	MSE	Correlation Accuracy
Linear Regression	0.1037821	0.926
Linear Kernel Support vector machine	0.097984	0.934
Radial Kernel Support Vector Machine	0.1843159	0.873
K-Nearest neighbors (KNN)	0.1349	0.902

All the models perform relatively well on our test set. We can see that Linear Kernel SVM performs best with an MSE of 0.0978 which is lowest among all and a correlation accuracy of 0.934 which is highest among all the algorithms tested. It tells how well the model handles our balanced dataset.

Conclusion

We wanted to determine which algorithm predicts the per day number of Covid-19 cases in the USA. We can conclude that with our given balanced and scaled dataset, Linear Kernel SVM performs best with the minimum error and highest accuracy. If we get similar data, we can rely on it for giving us the most accurate results. But with the advancement in the vaccine research, we need to keep an eye on it and add that factor in the data else the performance of the model will be affected. The Linear Regression model performs the second best with very less difference in the MSE. It shows the relationship of variables with our response variable.