

INTRODUCTION

Coronavirus(COVID-19) has become the most buzzed topic these days. COVID-19 is the disease caused by the new coronavirus that emerged in China in December 2019. The source of this virus is believed to be a 'wet market' in Wuhan which sold both dead and live animals including fishes and birds.novel coronavirus has inflicted havoc across the globe for lives and livelihoods. The impact of the pandemic on human lives is severe, but the effects on the global economy and on sustainable development's future are also a concern. The International Monetary Fund already declared that the world is into a recession. The full economic impact of the crisis is still difficult to predict but preliminary estimates are US\$2 trillion.

COVID-19 symptoms include cough, fever, shortness of breath,dry cough, headache, pneumonia. COVID-19 can be severe and some cases have caused death.

We will analyze the outbreak of coronavirus across various regions, visualize them using charts and graphs and predict the number of upcoming cases for the next 10 days using linear regression and polynomial regression models in python. The data has information from 31th December 2019 to till.



Collecting data from kaggle

Kaggle is a platform for predictive modelling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users. This crowdsourcing approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be most effective. On 8 March 2017. Google announced that they were acquiring Kaggle. They will join the Google Cloud team and continue to be a distinct brand. In January 2018, Booz Allen and Kaggle launched Data Science Bowl, a machine learning competition to analyze cell images and identify nuclei.

Methodology

Implementing Polynomial Regression With scikit-learn

Implementing polynomial regression with scikit-learn is very similar to linear regression. There is only one extra step: you need to transform the array of inputs to include non-linear terms such as x^2 .

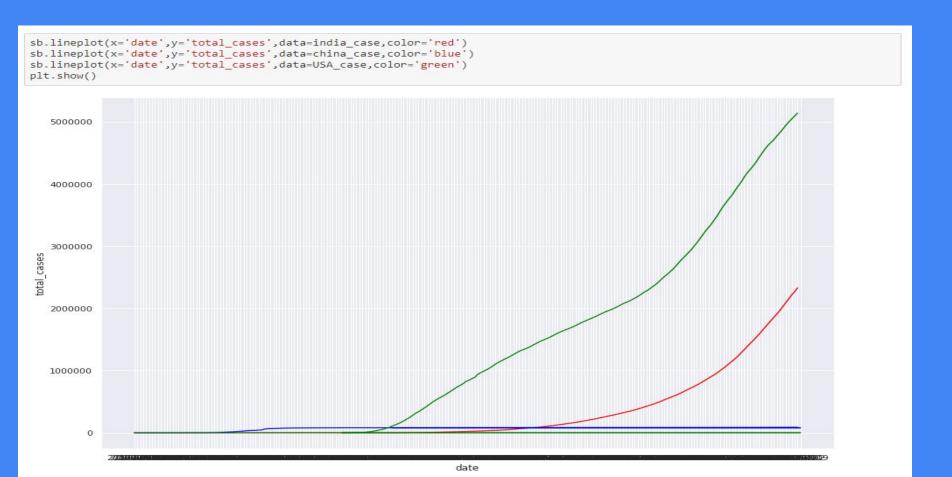
Degree is an integer (2 by default) that represents the degree of the polynomial regression function. In our project, we have taken the degree 3 of polynomial regression for better accuracy.

The result came out through this model is very good and has very good accuracy of 98.7.

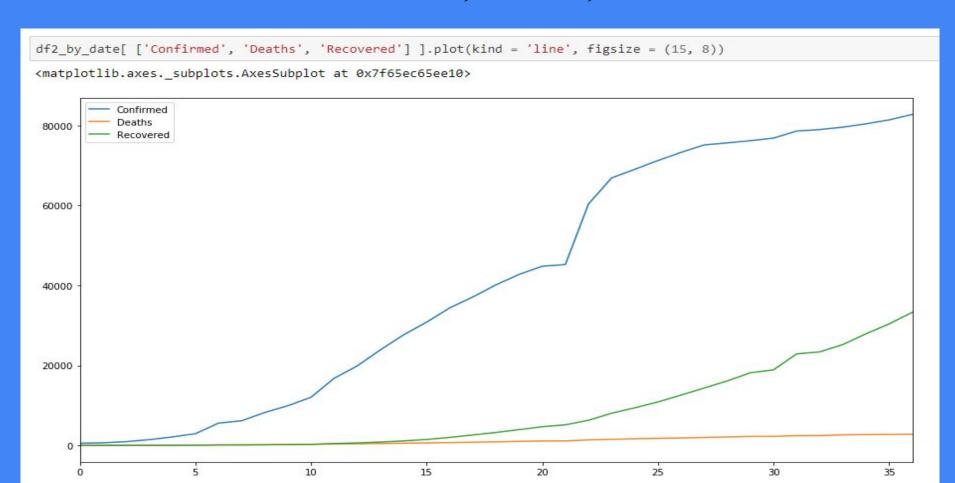
Countries affected by covid-19



Comparison of cases between India, China and US



Visualization Confirmed, Deaths, Recovered Cases



Heatmap representation (checking correlation)

sb.heatmap(india case.corr().abs(),annot=True)

<matplotlib.axes._subplots.AxesSubplot at 0x7effd0131748>

total cases new cases total deaths new deaths total cases per million new cases per million total deaths per million new deaths per million total tests total tests per thousand new tests_per_thousand new tests smoothed new tests smoothed per thousand tests per case positive rate stringency index population population density median age aged 65 older aged 70 older gdp per capita extreme poverty cardiovasc death rate diabetes prevalence female smokers male smokers handwashing facilities hospital beds per thousand life expectancy

87 1 0.990.990.870.960.990.990.970.990.990.610.850.31 0.790.99 1 0.990.890.99 1 0.990.890.970.990.990.970.980.98 9 c2 1459 P.9 c 9 65 T58- 4 64- 5 74- 7 99 - 7 99 - 149 - 158- 149 - 149 -0.810.990.99 1 0.90.990.99 1 0.90.97 1 1 0.970.980.980.7 0.910.35 d-158 158-762 119-254 252 268-258-154 258-16-15 1 0.870.890.9 1 0.880.90.860.820.850.850.60.840.37 64-B52-P53-568-152-B67-552-16-84-5118-3-553-863-B52 1 0 940 940 87 1 0 940 940 870 940 940 940 970 940 940 60 850 81 3 e 5 e 5 l 6 2 P. 6 P. 6 P. 6 P. 11 6 F. 6 - 7 e 7 F. 6 - 11 6 e - 5 e 6 F. 6 -0.790.99 1 0.990.890.99 1 0.990.890.970.990.990.970.980.980.880.3 7 e4 58e-11e61 50 e2 186- 761- 162- 566- 162- 661- 164- 564- 566 e-7156-316-716-716-9156-3172-562-962-963-162-962-96 4-165 Nat-567 B68-965 B61-565 561-165 B61-171-16 087096097097088096097097088 1 0 980 97 1 0 980 98 0.820.990.99 1 0.90.990.99 1 0.90.98 1 1 0.970.990.99 0.920.990.99 1 0.860.990.99 1 0.860.97 1 1 @- B7 - 962 B6-0.890.970.970.970.820.970.970.82 1 0.970.97 1 0.980 980 60 840 45 6e**∃e716e8.17**e 3e-3e71.6e-3e711.8e-3e611.8e1-1.6e 0 0126-7661 Re-TIESTI Se-TIEST 5-4-1184-1184-1184-1184 0.920.990.980.980.850.990.980.980.850.980.990.990.98 4-6-11-2-13-15-15-16-65 P.G. P.G. P.G. VG4 VIG4 VIG4 VIG 7 g- 1465- 1 g- 1465- 1568- 15 4e-Test 9 & 1@-5@-5@-1@-5@-1@-1@-9@-5@-5@-5@ 8el-Val-9al-Val-5a-5a-76-21-81.6el-161-765-159-52-177.4e5-828-848-846-8 2. 1 22. 1 23. 1 23. 1 25. 5 5 4 6 2. 1 23. 1 24. 1 25. 8e-1e31 2e8 552 962 75-1e61 6e5 166 166 561 959 564 772 865 866 162 75 8ct 72t 92t 72t 52t 52t 76 2ts1 6ct 16t 72t 53t 16 156 57 46t 46t 15 8 2 2 26 5 62 1468 152 262 16 216 16 2 16 16 16 16 16 16 1 8.44 1438 1469- 1772- 1869- 1566 1768 1168- 1774- 1762- 1762- 1763- 1864- 1764- 1764- 1764- 1764- 1764- 1764-8*0*-73-59-59-79-54-44-163165-86-74-75-85-36-18-6-5*4*-44-46-15-8 (2. 1) 28 - 196 3 44. V 23. D 23. B 25. S 26 - 1 6 02. 9 45. D 24. D 25. D 25. D 25. D 25. D 26. D 26. D 27. D 27. D 27. D 27. 8 2 . 4 22 . 4 23 . 1 23 . 8 2 . 5 . 5 . 4 6 2 . 6 2 . 6 2 . 6 2 . 6 2 . 6 2 . 6 . 7 25 . 8 2 . 9 2 . 1 6 . 5 2 . 6 2 . 7 2 5 . 6 2 . 7 2 5 . 6 2 . 7 2 5 . 6 2 . 7 2 5 . 8¢-42-49-19-86-9546¢-84-16-56-56-99-16-46-49-86-56-56-56 er_thousand thousand

-1.0

- 0.8

- 0.6

-04

-02

