

Actualización de datos de Covid-19 en la infección hasta el 2021. ¶

Covid-19 infección en Ecuador. Modelos matemáticos y predicciones

Una comparación de modelos, lineal, polinómico, logísticos y exponenciales aplicados a la infección por el virus Covid-19

Se realiza un análisis matemático simple del crecimiento de la infección en Python y dos modelos para comprender mejor la evolución de la infección.

Se crean modelos de series temporales del número total de personas infectadas hasta la fecha (es decir, las personas realmente infectadas más las personas que han sido infectadas). Estos modelos tienen parámetros, que se estimarán por ajuste de curva.

In [1]:

```
1 # Importar las librerías para el análisis
2 import pandas as pd
3 import numpy as np
4 from datetime import datetime, timedelta
5 from sklearn.metrics import mean_squared_error
6 from scipy.optimize import curve_fit
7 from scipy.optimize import fsolve
8 from sklearn import linear_model
9 import matplotlib.pyplot as plt
10 %matplotlib inline
11
```

Url con datos actualizados al 26 de abril del 2021

In [2]:

```
1
2 url = 'http://cowid.netlify.com/data/owid-covid-data.csv'
3
4 df = pd.read_csv(url)
5 df
```

Out[2]:

| | iso_code | continent | location | date | total_cases | new_cases | new_cases_smoothed | 1 |
|-------|----------|-----------|-------------|------------|-------------|-----------|--------------------|---|
| 0 | AFG | Asia | Afghanistan | 2020-02-24 | 1.0 | 1.0 | NaN | |
| 1 | AFG | Asia | Afghanistan | 2020-02-25 | 1.0 | 0.0 | NaN | |
| 2 | AFG | Asia | Afghanistan | 2020-02-26 | 1.0 | 0.0 | NaN | |
| 3 | AFG | Asia | Afghanistan | 2020-02-27 | 1.0 | 0.0 | NaN | |
| 4 | AFG | Asia | Afghanistan | 2020-02-28 | 1.0 | 0.0 | NaN | |
| ... | ... | ... | ... | ... | ... | ... | ... | |
| 85785 | ZWE | Africa | Zimbabwe | 2021-04-29 | 38235.0 | 44.0 | 31.000 | |
| 85786 | ZWE | Africa | Zimbabwe | 2021-04-30 | 38257.0 | 22.0 | 30.286 | |
| 85787 | ZWE | Africa | Zimbabwe | 2021-05-01 | 38260.0 | 3.0 | 28.000 | |
| 85788 | ZWE | Africa | Zimbabwe | 2021-05-02 | 38281.0 | 21.0 | 27.857 | |
| 85789 | ZWE | Africa | Zimbabwe | 2021-05-03 | 38293.0 | 12.0 | 27.286 | |

85790 rows × 59 columns

Imprimos los resultados y agregamos el numero del dia

In [3]:

```
1 df = df[df['location'].isin(['Ecuador'])] #Filtro la Informacion solo para Ecuador
2 df = df.loc[:,['date','total_cases']] #Selecciono las columnas de analisis
3 # Expresar las fechas en numero de dias desde el 01 Enero
4 FMT = '%Y-%m-%d'
5 date = df['date']
6 df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-0
7
8 df
```

Out[3]:

| | date | total_cases |
|-------|------|-------------|
| 22630 | 60 | 6.0 |
| 22631 | 61 | 6.0 |
| 22632 | 62 | 7.0 |
| 22633 | 63 | 10.0 |
| 22634 | 64 | 13.0 |
| ... | ... | ... |
| 23054 | 484 | 380689.0 |
| 23055 | 485 | 381862.0 |
| 23056 | 486 | 384589.0 |
| 23057 | 487 | 387299.0 |
| 23058 | 488 | 388046.0 |

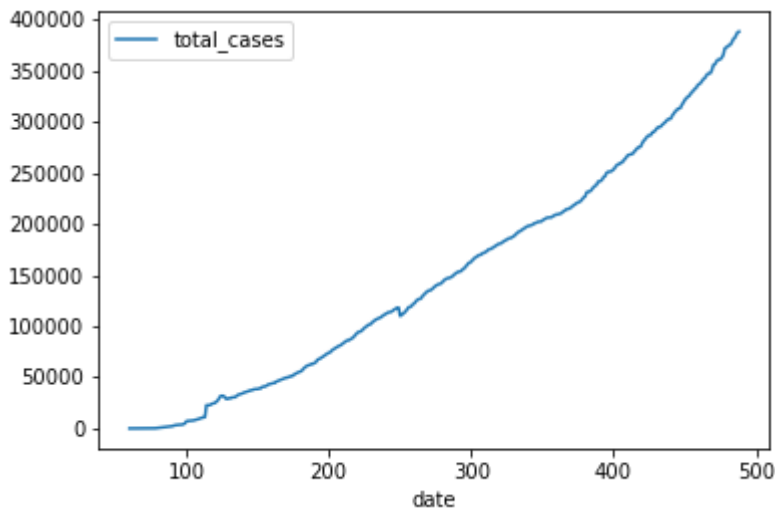
429 rows × 2 columns

In [4]:

```
1 df.plot(x='date', y='total_cases')
```

Out[4]:

<matplotlib.axes._subplots.AxesSubplot at 0x140e1b9e700>



Ahora podemos analizar los cuatro modelos que tomaré en el examen, que son la función lineal, polinómica, logística y la función exponencial. Cada modelo tiene tres parámetros, que se estimarán mediante un cálculo de ajuste de curva en los datos históricos.

EL modelo lineal

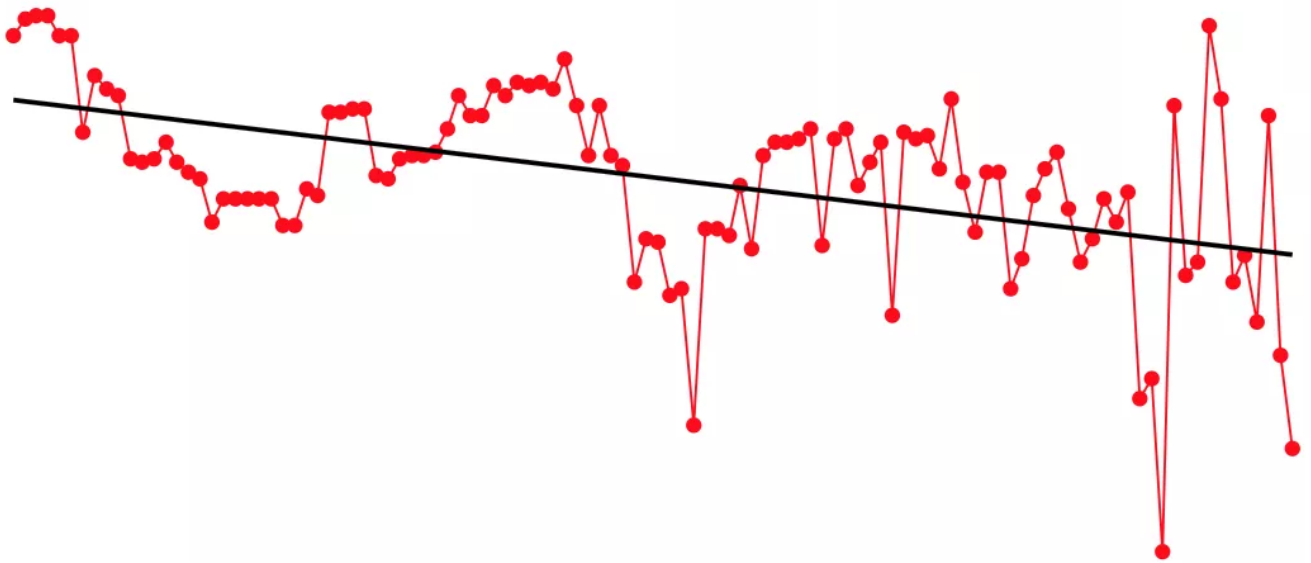
La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en Machine Learning y en estadística. En su versión más sencilla, lo que haremos es «dibujar una recta» que nos indicará la tendencia de un conjunto de datos continuos.

Recordemos rápidamente la fórmula de la recta:

$$Y = mX + b$$

Donde Y es el resultado, X es la variable, m la pendiente (o coeficiente) de la recta y b la constante o también conocida como el «punto de corte con el eje Y» en la gráfica (cuando X=0) Ejemplo

The development in Pizza prices in Denmark from 2009 to 2018



Recordemos que los algoritmos de Machine Learning Supervisados, aprenden por sí mismos y -en este caso- a obtener automáticamente esa «recta» que buscamos con la tendencia de predicción. Para hacerlo se mide el error con respecto a los puntos de entrada y el valor «Y» de salida real.

In [5]:

```
1 x = list(df.iloc[:, 0]) # Fecha
2 y = list(df.iloc[:, 1]) # Numero de casos
3 # Creamos el objeto de Regresión Lineal
4 regr = linear_model.LinearRegression()
5
6 # Entrenamos nuestro modelo
7 regr.fit(np.array(x).reshape(-1, 1), y)
8
9 # Veamos Los coeficientes obtenidos, En nuestro caso, serán La Tangente
10 print('Coefficients: \n', regr.coef_)
11 # Este es el valor donde corta el eje Y (en X=0)
12 print('Independent term: \n', regr.intercept_)
13 # Error Cuadrado Medio
```

```
Coefficients:
[881.23780419]
Independent term:
-91952.27722827051
```

De la ecuación de la recta $y = mX + b$ nuestra pendiente «m» es el coeficiente y el término independiente «b»

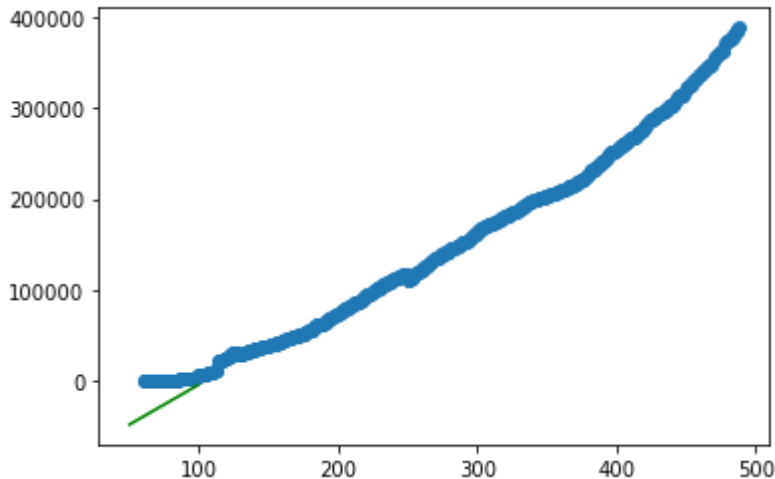
In [6]:

```
1 #Vamos a comprobar:
2 # Quiero predecir cuántos "Casos" voy a obtener por en el día 100,
3 # según nuestro modelo, hacemos:
4 y_prediccion = regr.predict([[100]])
5 print(int(y_prediccion))
```

In [7]:

```
1 #Graficar
2 plt.scatter(x, y)
3 x_real = np.array(range(50, 100))
4 print(x_real)
5 plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='green')
6 plt.show()
7
```

```
[50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
98 99]
```



El modelo logístico

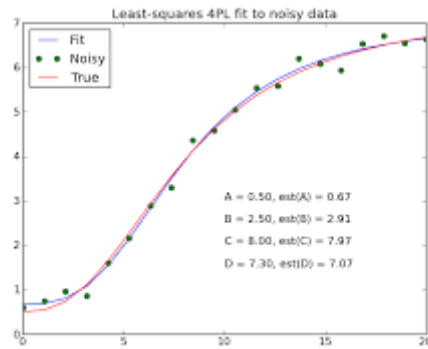
El modelo logístico se ha utilizado ampliamente para describir el crecimiento de una población. Una infección puede describirse como el crecimiento de la población de un agente patógeno, por lo que un modelo logístico parece razonable. La expresión más genérica de una función logística es:

$$f(x, a, b, c) = \frac{c}{1 + e^{-(x-b)/a}}$$

En esta fórmula, tenemos la variable x que es el tiempo y tres parámetros: a, b, c.

- a se refiere a la velocidad de infección
- b es el día en que ocurrieron las infecciones máximas
- c es el número total de personas infectadas registradas al final de la infección

A continuación se puede apreciar un ejemplo de regresión logística



Definamos la función en Python y realicemos el procedimiento de ajuste de curva utilizado para el crecimiento logístico.

In [8]:

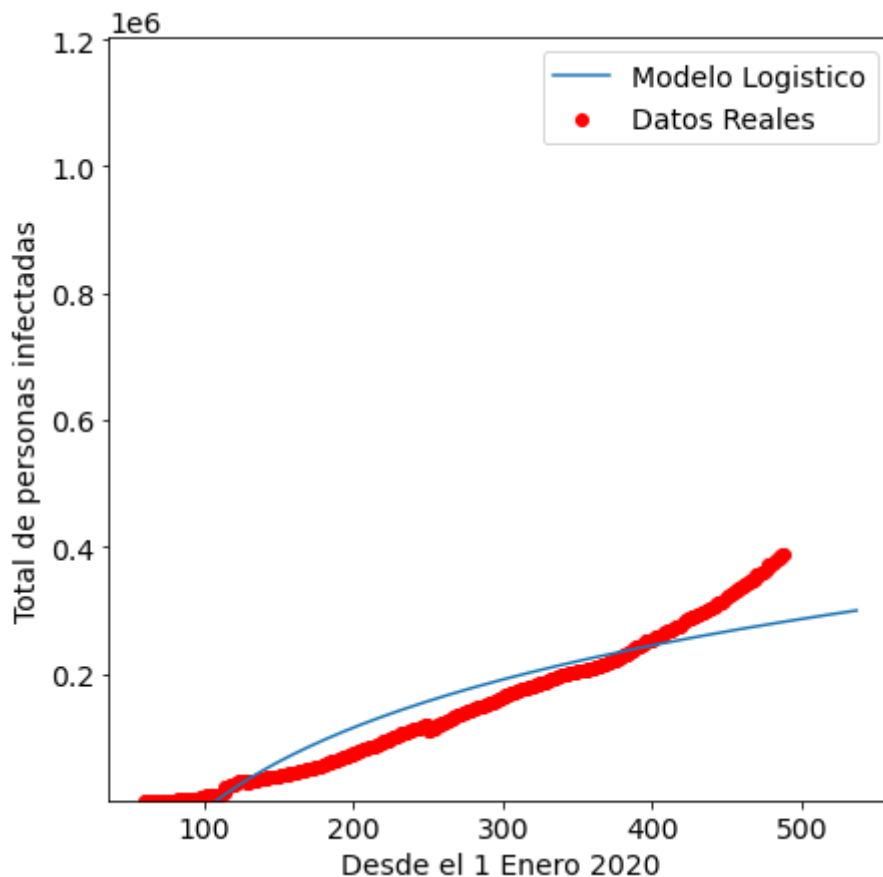
```
1 def modelo_logistico(x,a,b):
2     return a+b*np.log(x)
3
4 exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de los paramatros
5 print(exp_fit)
6
```

```
(array([-875046.93630738, 186848.37448837]), array([[ 3.98391454e+08, -7.19
371746e+07],
          [-7.19371746e+07, 1.31192173e+07]]))
```

Graficas

In [9]:

```
1 pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
2 plt.rcParams['figure.figsize'] = [7, 7]
3 plt.rc('font', size=14)
4 # Real data
5 plt.scatter(x,y,label="Datos Reales",color="red")
6 # Predicted exponential curve
7 plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], label="Modelo Logistico")
8 plt.legend()
9 plt.xlabel("Desde el 1 Enero 2020")
10 plt.ylabel("Total de personas infectadas")
11 plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
12 plt.show()
```



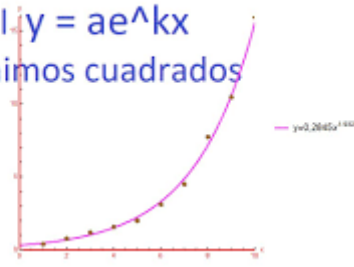
Modelo exponencial

Mientras que el modelo logístico describe un crecimiento de infección que se detendrá en el futuro, el modelo exponencial describe un crecimiento de infección imparabile. Por ejemplo, si un paciente infecta a 2 pacientes por día, después de 1 día tendremos 2 infecciones, 4 después de 2 días, 8 después de 3 y así sucesivamente.

$$f(x, a, b, c) = a \cdot e^{b(x-c)}$$

A continuacion se tiene un ejemplo de regresion exponencial

Curva de ajuste para una función tipo
exponencial $y = ae^{kx}$
usando mínimos cuadrados



In [10]:

```
1 # Implementar
```

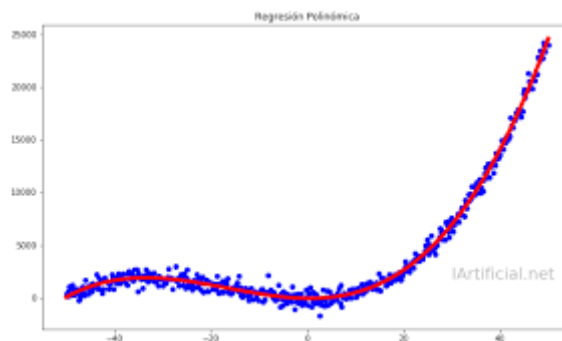
Modelo polinomial

Predicción de una variable de respuesta cuantitativa a partir de una variable predictora cuantitativa, donde la relación se modela como una función polinomial de orden n (esto significa que pueden tener de diferentes exponentes o grados y se debe ir probando)

Se puede tener una ecuación con diferentes grados

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n + \varepsilon$$

Ejemplo de una regresión polinómica de grado 4.



In [11]:

```
1 # Implementar
2
3
4 # Se puede implementar modelos adicionales, en caso de ser así explicar o dar una intro
5
6 # Se tomara como puntos adicionales al trabajo.
```

Analisis

Conclusiones

Criterio personal (politico, economico y social de la situacion)

Referencias

- https://www.researchgate.net/publication/340092755_Infeccion_del_Covid-19_en_Colombia_Una_comparacion_de_modelos_logisticos_y_exponenciales_aplicados_a_la_infeccion_por_SARS-CoV-2
- <https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>

In []:

1