

## 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 crea 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 [14]: 1 # Importar Las librerías para el analisis
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
```

```
In [15]: 1 # Actualizar Los datos (URL)
2
3 url = 'http://cowid.netlify.com/data/full_data.csv'
4
5 df = pd.read_csv(url)
6 df
```

```
Out[15]:
```

	date	location	new_cases	new_deaths	total_cases	total_deaths
0	2020-02-25	Afghanistan	NaN	NaN	1	NaN
1	2020-02-26	Afghanistan	0.0	NaN	1	NaN
2	2020-02-27	Afghanistan	0.0	NaN	1	NaN
3	2020-02-28	Afghanistan	0.0	NaN	1	NaN
4	2020-02-29	Afghanistan	0.0	NaN	1	NaN
...	...	...	...	...	...	...
2862	2020-03-13	World	7488.0	338.0	132758	4958.0
2863	2020-03-14	World	9781.0	433.0	142534	5392.0
2864	2020-03-15	World	10987.0	343.0	153517	5735.0
2865	2020-03-16	World	13971.0	855.0	167506	6608.0
2866	2020-03-17	World	11584.0	819.0	179112	7428.0

2867 rows x 6 columns

Imprimos los resultados y agregamos el numero del día

Imprimos los resultados y agregamos el numero del día

```
In [16]: 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 días desde el 01 Enero
4 FMT = '%Y-%m-%d'
5 date = df['date']
6 print(type(date))
7 df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01-01", FMT)).days)
8
9 df
```

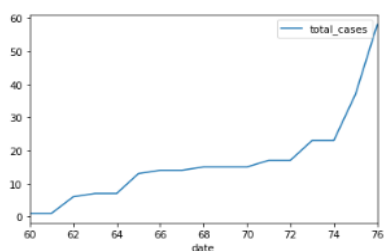
<class 'pandas.core.series.Series'>

```
Out[16]:
```

	date	total_cases
681	60	1
682	61	1
683	62	6
684	63	7
685	64	7
686	65	13
687	66	14
688	67	14
689	68	15
690	69	15
691	70	15
692	71	17
693	72	17
694	73	23
695	74	23
696	75	37
697	76	58

```
In [30]: 1 df.plot(x='date', y='total_cases')
```

Out[30]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7fcfb7f2e710>



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

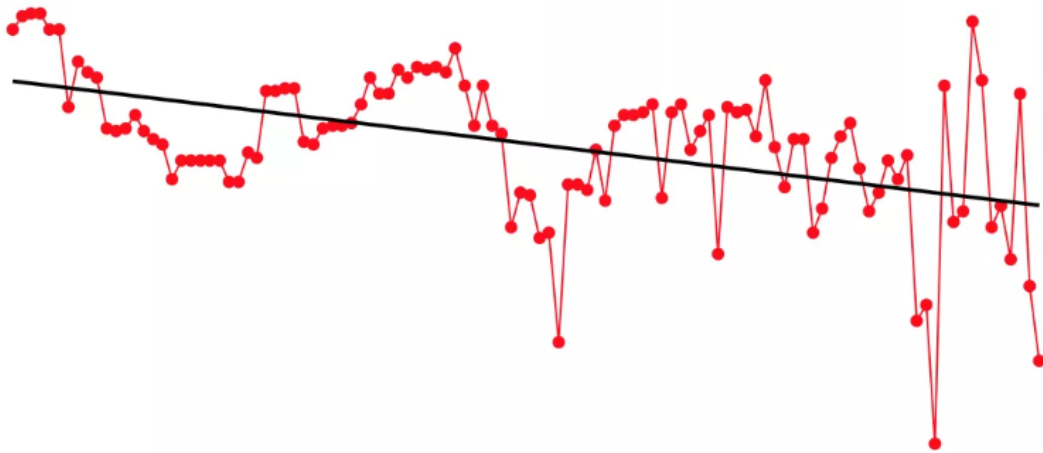
La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en Machine Learning y en estadística. En su versión más sencilla, lo que haremos es «dibujar una recta» que nos indicará la tendencia de un conjunto de datos continuos.

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 [14]: 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('Coeficientes: \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

Coeficientes:
[2.31617647]
Independent term:
-140.85294117647058
```

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

```
In [16]: 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))

90
```

```
In [21]: 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

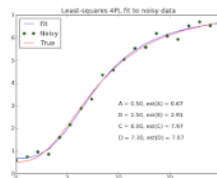
El modelo logístico se ha utilizado ampliamente para describir el crecimiento de una población. Una infección puede describirse como el crecimiento de la población de un agente patógeno, por lo que un modelo logístico parece razonable. La expresión más genérica de una función logística es:

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

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



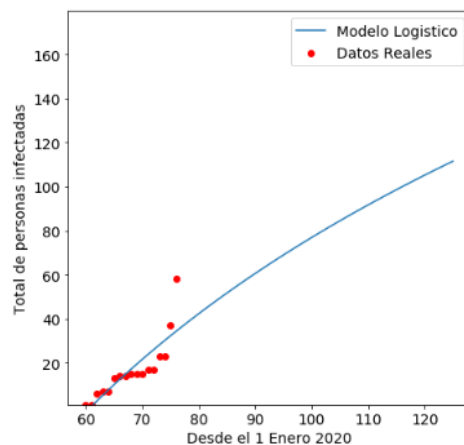
Definamos la función en Python y realicemos el procedimiento de ajuste de curva utilizado para el crecimiento logístico.

```
In [31]: 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 parametros
5 print(exp_fit)
6
```

```
(array([-637.69367899, 155.1710206 ]), array([[11872.54041468, -2814.63865054],
       [-2814.63865054, 667.4662425 ]]))
```

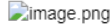
## Graficas

```
In [34]: 1 pred_x = list(range(min(x),max(x)+50)) # Predecir 50 días 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 límites de Y
12 plt.show()
```

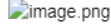


## Modelo exponencial

Mientras que el modelo logístico describe un crecimiento de infección que se detendrá en el futuro, el modelo exponencial describe un crecimiento de infección impararable . 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.

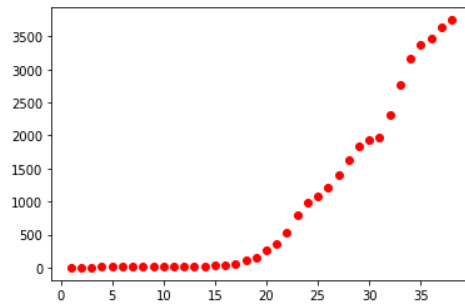


A continuacion se tiene un ejemplo de regresion exponencial



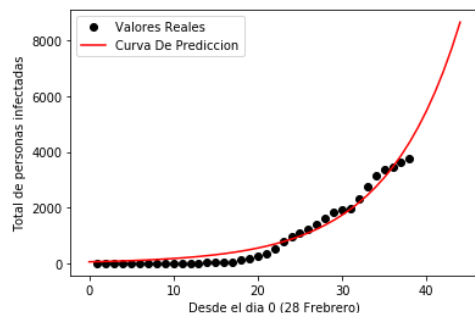
```
In [5]: #datos actualizados desde la pagina wikipedia
#https://es.wikipedia.org/wiki/Pandemia_de_enfermedad_por_coronavirus_de_2020_en_Ecuador
import matplotlib.pyplot as plt
from pylab import *
from scipy.stats import expon
matrizdatos=[(1,1),(2,6),(3,6),(4,11),(5,14),(6,14),(7,14),(8,14),(9,15),(10,15),(11,17),(12,17),(13,17),
(14,23),(15,28)
,(16,37),(17,58),(18,111),(19,155),(20,260),(21,367),(22,532),(23,789),(24,981),(25,1082),(26,1211),(27,14
03),(28,1627)
,(29,1835),(30,1924),(31,1966),(32,2302),(33,2758),(34,3163),(35,3368),(36,3465),(37,3646),(38,3747)]
x_real= np.zeros(38)
y_real= np.zeros(38)
for i in range(38):
    x_real[i]=matrizdatos[i][0]
    y_real[i]=matrizdatos[i][1]

plt.scatter(x_real,y_real,label="Datos Reales",color="red")
show()
```



```
In [7]: from scipy.optimize import curve_fit
def func(x, a, b):
    return a * np.exp(-b * x-0)
r=curve_fit(func, x_real, y_real)
print(r)
x_prec=np.array(range(0,45))
r1=func(42,r[0][0],r[0][1])
print(r1)
plt.figure()
plt.plot(x_real, y_real, 'ko', label="Valores Reales")
plt.plot(x_prec,[func(i,r[0][0],r[0][1]) for i in x_prec], 'r-', label="Curva De Prediccion")
plt.legend()
plt.xlabel("Desde el dia 0 (28 Febrero)")
plt.ylabel("Total de personas infectadas")
plt.show()
```

```
(array([55.22188039, -0.11486961]), array([[1.17279857e+02, 6.12481128e-02],
[6.12481128e-02, 3.24975488e-05]]))
6876.62357323644
```



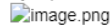
## Modelo polinomial

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 exponenciales 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.

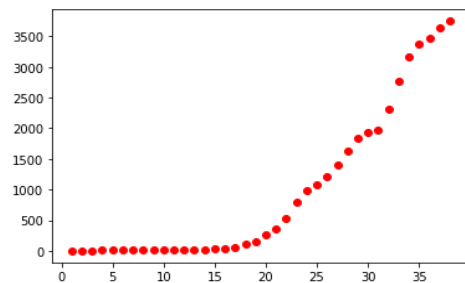


```
In [8]: #datos actualizados desde La pagina wikipedia
#https://es.wikipedia.org/wiki/Pandemia_de_enfermedad_por_coronavirus_de_2020_en_Ecuador
import matplotlib.pyplot as plt
from pylab import *
from scipy.stats import expon
matrizdatos=[(1,1),(2,6),(3,6),(4,11),(5,14),(6,14),(7,14),(8,14),(9,15),(10,15),(11,17),(12,17),(13,17),
(14,23),(15,28)
,(16,37),(17,58),(18,111),(19,155),(20,260),(21,367),(22,532),(23,789),(24,981),(25,1082),(26,1211),(27,14
03),(28,1627)
,(29,1835),(30,1924),(31,1966),(32,2302),(33,2758),(34,3163),(35,3368),(36,3465),(37,3646),(38,3747)]
x_real= np.zeros(38)
y_real= np.zeros(38)
for i in range(38):
    x_real[i]=matrizdatos[i][0]
    y_real[i]=matrizdatos[i][1]

plt.scatter(x_real,y_real,label="Datos Reales",color="red")
show()

# Se puede implementar modelos adicionales, en caso de ser así explicar o dar una introducción del modelo
aplicado

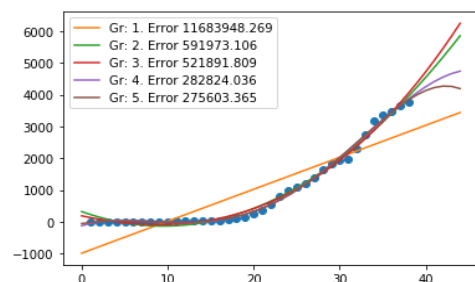
# Se tomara como puntos adicionales al trabajo.
```



```
In [9]: # Calcular ajustes para diferentes grados
import numpy as np
import matplotlib.pyplot as plt
sols = {}
for grado in range(1,6):
    z = np.polyfit(x_real, y_real, grado, full=True)
    sols[grado] = z
xp = np.array(range(0,45))
for grado, sol in sols.items():
    coefs, error, *_ = sol
    p = np.polyid(coefs)
    print(p(42))
    plt.plot(xp, p(xp), "--", label="Gr: %s. Error %.3f" % (grado, error) )
plt.legend()

3235.733887733889
5176.717155414519
5448.253792154952
4550.610153292224
4267.442925299156
```

Out[9]: <matplotlib.legend.Legend at 0x1354ca30e08>



## Analisis

Teniendo ahora los dos modelos los compararemos con los datos oficiales al día 10/04/2020 nos dice que existe 7161 casos confirmados en la fuente de información([https://es.wikipedia.org/wiki/Pandemia\\_de\\_enfermedad\\_por\\_coronavirus\\_de\\_2020\\_en\\_Ecuador](https://es.wikipedia.org/wiki/Pandemia_de_enfermedad_por_coronavirus_de_2020_en_Ecuador)) El modelo de exponencial nos dice que para esta fecha existen 6876.62357323644 lo cual nos da un margen de error de 284.38 infectados El modelo polinomial nos dice que para esta fecha en un polonomio de tercer grado existen 5448.253792154952 este nos da un margen de error de 1.712,75 infectados

## Conclusiones

Con los resultados que hemos obtenido podemos decir que el modelo exponencial es mejor para poder predecir la curva de infectados en el territorio nacional debido a que entre los modelos que el exponencial tiene un grado menor de error no obstante el modelo polinomial es el que se adapta de mejor manera a los casos reales pero se desliza mucho de la predicción

Las librerías usadas han sido muy prácticas debo recalcar que son fáciles de usar y nos un gran margen de uso para diferentes modelos de predicción en caso de tener algún inconveniente o algo parecido podemos consultar la documentación para un mejor uso

## Criterio personal (político, económico y social de la situación)

Política: con respecto a la parte política en el país hay un descontento con respecto a nuestra máxima autoridad que el presidente debido al poco interés de su parte esto genera que las personas se sientan un poco desprotegidas por parte de su mandatario y dado a lugar que nuevas figuras políticas sobresalgan a la luz así como el vicepresidente, lo que sí ha sido evidente lo poco y mal preparados que estamos para este tipo de emergencias

Económica: la parte económica será una de las más afectadas debido a la paralización de las fábricas como la falta de turismo exportaciones y demás actos económicos, esto era un duro golpe para el país esperamos que al final de emergencia todo se normalice a nivel nacional

Social: en el ámbito social se tiene el país tiene que aprender a tener una costumbre de comunicación real debido a la falta de información en la provincia del Guayas a la falta de conocimiento en foco de epidemia se les ha ido de las manos y las personas siguen sin entender lo cual es lamentable socialmente el país debe mejorar un montón

## 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\\_el\\_virus\\_en\\_Colombia](https://www.researchgate.net/publication/340092755_Infeccion_del_Covid-19_en_Colombia_Una_comparacion_de_modelos_logisticos_y_exponenciales_aplicados_a_la_infeccion_por_el_virus_en_Colombia)
- <https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>
- [https://es.wikipedia.org/wiki/Pandemia\\_de\\_enfermedad\\_por\\_coronavirus\\_de\\_2020\\_en\\_Ecuador](https://es.wikipedia.org/wiki/Pandemia_de_enfermedad_por_coronavirus_de_2020_en_Ecuador)