

# MODELO LINEAL

**AUTOR: CARLOS MOROCHO**

## **1) Formulación del problema**

En este proyecto se aplicará el modelo de regresión logística para realizar la predicción de casos nuevos confirmados de COVID-19 en el Ecuador. Para este propósito se utilizarán como variables primordiales el número de casos confirmados y la fecha en formato de días. Además se realizará una comparación con el modelo lineal para ver cual se ajusta mejor a los datos analizados.

## **2) Definición del sistema**

Para realizar la simulación se utilizará un cuaderno de Python (Jupyter Notebook), en el que el usuario podrá ir ejecutando segmentos de código y ver como se desarrolla el entrenamiento y predicción del modelo.

## **3) Formulación del modelo**

Como se mencionó previamente, el modelo a utilizar en este proyecto será el de regresión logística.

## **4) Colección de datos**

Los datos a utilizar son los de los informes nacionales del Servicio Nacional de Gestión de Riesgos y Emergencias del Ecuador (SNGRE), Ministerio de Salud Pública del Ecuador (MSP), y Registro Civil del Ecuador.

## 5) Implementación

```
In [1]: #Importación de las librerías necesarias
import pandas as pd
import numpy as np
from datetime import datetime
from sklearn import linear_model
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: #Obtención de los datos de COVID-19 de Ecuador
url_datos = 'DatosCOVID2020.csv'

datos = pd.read_csv(url_datos, sep = ',')
datos
```

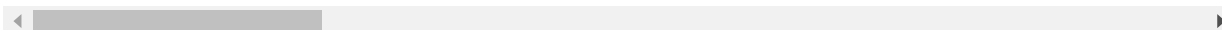
Out[2]:

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muertes_confirmadas	m
0	129	129	0	106	1	
1	206	206	77	178	2	
2	273	273	67	236	2	
3	354	354	81	296	2	
4	762	762	408	651	2	
5	762	762	0	282	3	
6	1153	1153	391	481	4	
7	1670	1670	517	711	7	
8	2051	2051	381	870	7	
9	2360	2360	309	699	14	
10	2780	2780	420	708	18	
11	3618	3618	838	1311	27	
12	4290	4290	672	1692	29	

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muertes_confirmadas	m
13	5090	5090	800	1965	34	
14	5915	5915	825	2347	41	
15	6615	6615	700	2680	48	
16	6992	6992	377	2869	58	
17	7451	7451	459	3232	62	
18	8251	8251	800	3423	79	
19	9019	9019	768	3428	98	
20	9604	9604	585	3302	120	
21	10317	10317	713	3661	145	
22	11309	11309	992	4475	172	
23	12386	12386	1077	5137	180	
24	13039	13039	653	5449	191	
25	14406	14406	1367	6455	220	
26	15526	15526	1120	6868	242	
27	19102	19102	3576	9463	272	
28	21568	21568	2466	7911	297	
29	22649	22649	1081	8448	315	
...	...	...	...	...	...	
217	507559	486987	5798	69369	8072	
218	512457	491885	4898	71367	8087	
219	515705	495133	3248	70258	8099	
220	516424	495852	719	70411	8106	
221	520243	499671	3819	72099	8115	
222	525803	505231	5560	71511	8160	
223	530086	509514	4283	71121	8195	

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muertes_confirmadas	m
224	535429	514857	5343	67043	8221	
225	537556	516984	2127	61807	8235	
226	542463	521891	4907	56590	8248	
227	543529	522957	1066	56235	8266	
228	555016	534444	11487	63165	8280	
229	557421	536849	2405	59355	8297	
230	558365	537793	944	53342	8312	
231	562074	541502	3709	53605	8321	
232	567080	546508	5006	53176	8357	
233	569362	548790	2282	51707	8371	
234	569798	549226	436	50815	8380	
235	570515	549943	717	49712	8386	
236	577335	556763	6820	51224	8394	
237	579156	558584	1821	50370	8420	
238	583731	563159	4575	50582	8449	
239	587193	566621	3462	49940	8492	
240	593170	572598	5977	49175	8507	
241	596540	575968	3370	50989	8516	
242	597099	576527	559	49598	8525	
243	600741	580169	3642	49067	8592	
244	605331	584759	4590	50256	8614	
245	610265	589693	4934	50604	8642	
246	614531	593959	4266	50485	8658	

247 rows × 32 columns



```
In [3]: #Se filtran los datos para obtener solo los casos positivos
datos_infectados = datos.loc[:, ['positivas', 'created_at']]

#Expresar los datos de fecha en número de días desde el inicio de año
formato = '%d/%m/%Y'
fecha = datos_infectados['created_at']
datos_infectados['created_at'] = fecha.map(lambda x:
    (datetime.strptime(x, formato) - datetime.strptime('01/01/2020', fo
rmato)).days
)

datos_infectados
```

Out[3]:

	positivas	created_at
0	23	72
1	28	73
2	37	74
3	58	75
4	111	76
5	168	77
6	260	78
7	426	79
8	532	80
9	789	81
10	981	82
11	1082	83
12	1211	84
13	1403	85
14	1627	86

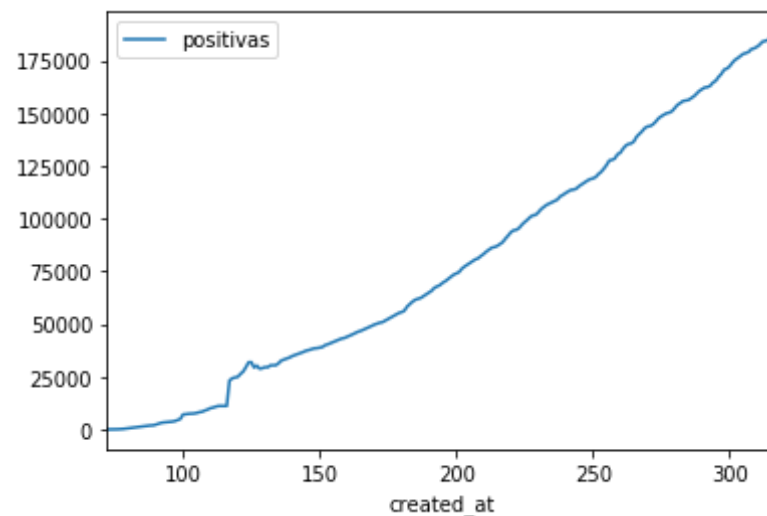
	positivas	created_at
15	1835	87
16	1924	88
17	1966	89
18	2302	90
19	2758	91
20	3163	92
21	3368	93
22	3465	94
23	3646	95
24	3747	96
25	3995	97
26	4450	98
27	4965	99
28	7161	100
29	7257	101
...	...	...
217	160615	289
218	161378	290
219	162245	291
220	162379	292
221	163071	293
222	164581	294
223	165407	295
224	167226	296
225	168570	297

	positivas	created_at
226	170591	298
227	171134	299
228	172148	300
229	173864	301
230	175258	302
231	176103	303
232	177148	304
233	178150	305
234	178518	306
235	179066	307
236	180389	308
237	180739	309
238	181464	310
239	182442	311
240	183863	312
241	184225	313
242	184667	314
243	185586	315
244	186469	316
245	187630	317
246	188583	318

247 rows × 2 columns

```
In [4]: datos_infectados.plot(x = 'created_at', y = 'positivas')
```

Out[4]: <matplotlib.axes.\_subplots.AxesSubplot at 0x262d95dd0f0>



Como se puede observar, el número de casos inicia el día 72 (13/03/2020) con 23 casos confirmados. Hasta el día 318 (14/11/2020) se registran 188583 casos confirmados.

```
In [5]: #Definición del modelo logístico
modelo_logistico = lambda x, a, b: a + b * np.log(x)

#Obtención de las variables para el entrenamiento
x = list(datos_infectados.iloc[:, 1]) #Fecha (Número de día)
y = list(datos_infectados.iloc[:, 0]) #Numero de infectados

#Realizamos el ajuste de curva para los datos
ajuste = curve_fit(modelo_logistico, x, y)
```

## 6) Verificación

La verificación de la implementación del modelo puede ser realizada mediante la ejecución paso a paso del propio cuaderno de python. En este caso no se han detectado errores, por lo que se prosigue al diseño de experimentos



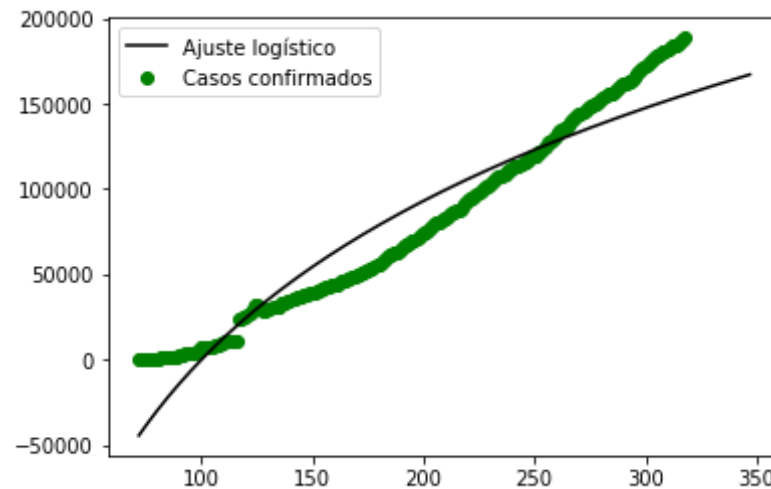
## 7) Diseño de experimentos

Para probar el modelo se utilizara como premisa un periodo de predicción de 30 días posteriores a la fecha 14/11/2020. Al ser un modelo sencillo unicamente se realizara una simulación.

## 8) Experimentación

```
In [8]: #Comprobación de la predicción para 30 días despues del 14/11/2020
dias = 30
x_real = list(range(min(x), max(x) + dias)) #Realizamos la predicción p
ara un mes

plt.scatter(x, y, color = 'green')
plt.plot(x_real, [modelo_logistico(i, ajuste[0][0], ajuste[0][1]) for i
in x_real], color = 'black')
plt.legend(('Ajuste logístico', 'Casos confirmados'))
plt.show()
```



## 9) Interpretación

De acuerdo a los parámetros de entrada de casos confirmados y fecha, se puede observar que el modelo no se ajusta correctamente los datos, por lo que sería conveniente buscar y utilizar otro tipo de función para predecir.

## 10) Implementación

Este proyecto no cuenta con interfaz gráfica de usuario externa, sin embargo, el usuario que desee ver el comportamiento del sistema cambiando el número de días a predecir lo puede hacer modificando la variable 'dias' ubicada en `ln[25]` de este cuaderno (Para el modelo logístico).

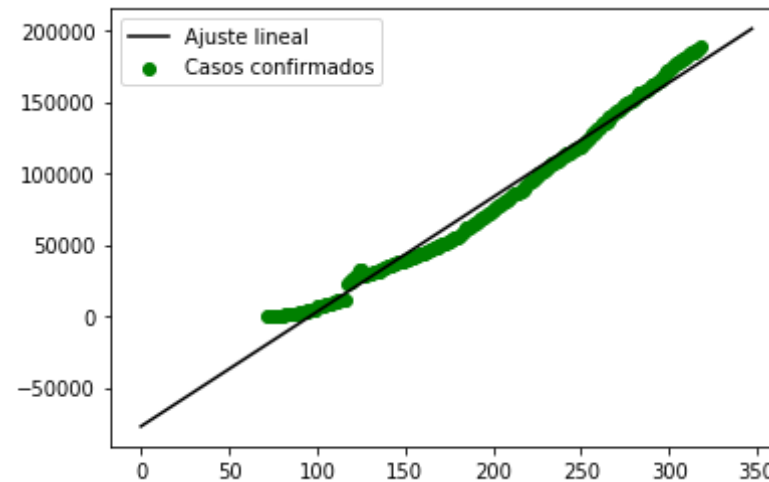
## 11) Uso del modelo lineal para el mismo conjunto de datos

```
In [7]: #Creación y entranamiento del modelo
modelo_lineal = linear_model.LinearRegression()
modelo_lineal.fit(np.array(x).reshape(-1, 1), y)

#Comprobación de la predicción para 30 días despues del 14/11/2020
dia = x[-1] + 30 #Se obtiene el dia a predecir sumando la ultima fecha
registrada mas un mes
prediccion_semana = modelo_lineal.predict([[dia]])

#Gráfica de los datos y la predicción
x_real = np.array(range(0, 348))

plt.scatter(x, y , color = 'green')
plt.plot(x_real, modelo_lineal.predict(x_real.reshape(-1, 1)), color =
'black')
plt.legend(('Ajuste lineal', 'Casos confirmados'))
plt.show()
```



En este caso se muestra la reacción del modelo lineal ante el conjunto de datos.

## 12) Análisis

El modelo logístico utilizado para la muestra de datos presenta una curva que no se ajusta adecuadamente a esto, por lo que la predicción de casos en el futuro se ve equivocada por un margen de error considerable. Por el contrario al usar el modelo lineal se puede observar que si bien podría mejorar, el ajuste es mejor que el del modelo logístico (Para esta muestra de datos).

## 13) Conclusiones

Mediante el presente trabajo se establece que de acuerdo a la comparativa realizada entre los modelos logístico y lineal, el lineal es claramente mejor cuando el conjunto de muestras tiene una tendencia lineal y constante, mientras que el modelo logístico se recomienda para datos que presenten un gran crecimiento inicial y luego se vaya estabilizando con crecimiento moderado.

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

Debido a todo el caos que provoco el virus en todo el mundo, estos modelos de prediccon que si bien no son del todo exactos pueden ayudarnos a solventar algunas respuestas que todo la sociedad esta buscando, existen millones de empresas que de seguro estan desarrollando estos sistemas debido a la contingencia pero creo que ningunoo veneficia a la sociedad como la informacion como tal.



In [ ]: