



El modelo logístico - Pasos del Modelo

Materia:

Simulación

Docente:

Ing. Diego Quisi

Estudiante:

Ricardo Vinicio Jara Jara

```
In [44]: # Importar las librerias para el analisis
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model
import matplotlib.pyplot as plt
%matplotlib inline

#Mis Datos COVID EC

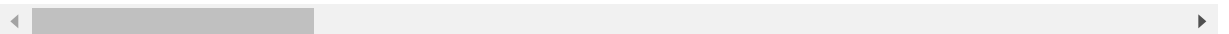
url = 'https://raw.githubusercontent.com/andrab/ecuacovid/master/datos_crudos/ecuacovid.csv'

df = pd.read_csv(url)
df
```

Out[44]:

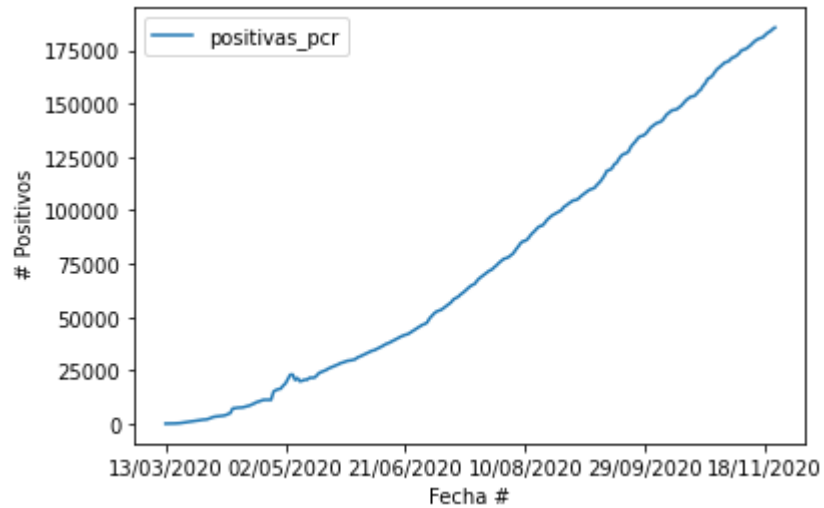
	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muertes_confirmadas	n
0	129	129	0	106		1
1	206	206	77	178		2
2	273	273	67	236		2
3	354	354	81	296		2
4	762	762	408	651		2
...
250	624030	603458	4335	48574		8708
251	629093	608521	5063	48617		8729
252	632453	611881	3360	48275		8750
253	636428	615856	3975	47265		8787
254	641542	620970	5114	45942		8804

255 rows × 32 columns



```
In [33]: ax = df.plot(x='created_at', y='positivas_pcr')
ax.set_xlabel("Fecha #")
ax.set_ylabel("# Positivos")

FMT = '%d/%m/%Y'
date = df['created_at']
df['created_at'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.s
trptime("01/01/2020", FMT)).days)
x = list(df.iloc[:, 0]) # Fecha
y = list(df.iloc[:, 1]) # Numero de casos
```



El modelo logístico

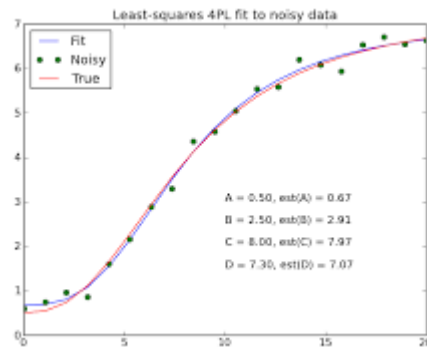
El modelo logístico se ha utilizado ampliamente para describir el crecimiento de una población. Una infección puede describirse como el crecimiento de la población de un agente patógeno, por lo que un modelo logístico parece razonable. La expresión más genérica de una función logística es:

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

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

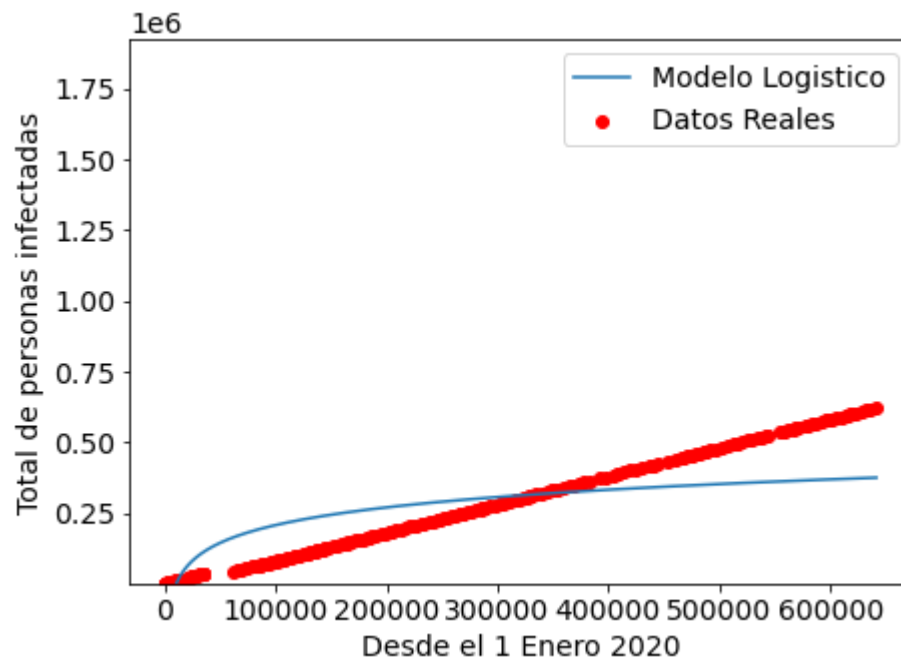
A continuación se puede apreciar un ejemplo de regresión logística



```
In [34]: def modelo_logistico(x,a,b):  
         return a+b*np.log(x)  
  
exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de los parametros  
print(exp_fit)  
  
(array([-829399.50124244,  90006.63182773]), array([[ 3.27437846e+09, -2.726  
28595e+08],  
          [-2.72628595e+08,  2.31133269e+07]]))
```

Grafica

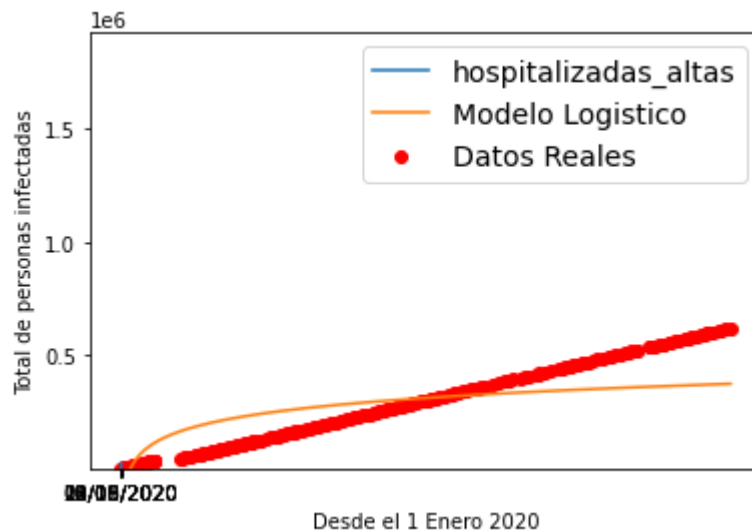
```
In [40]: pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
plt.rcParams['figure.figsize'] = [7, 5]
plt.rc('font', size=14)
# Real data
plt.scatter(x,y,label="Datos Reales",color="red")
# Predicted exponential curve
plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], label="Modelo Logistico" )
plt.legend()
plt.xlabel("Desde el 1 Enero 2020")
plt.ylabel("Total de personas infectadas")
plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir Los Limites de Y
plt.show()
```



Evaluar Recuperados

```
In [45]: df = pd.read_csv(url)
df.plot(x='created_at', y='hospitalizadas_altas')
FMT = '%d/%m/%Y'
date = df['created_at']
df['created_at'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.s
trptime("01/01/2020", FMT)).days)
x = list(df.iloc[:, 0]) # Fecha
y = list(df.iloc[:, 1]) # Numero de casos
exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de los parama
tros
print(exp_fit)
pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
plt.rcParams['figure.figsize'] = [7, 5]
plt.rc('font', size=14)
# Real data
plt.scatter(x,y,label="Datos Reales",color="red")
# Predicted exponential curve
plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pre
d_x], label="Modelo Logistico")
plt.legend()
plt.xlabel("Desde el 1 Enero 2020")
plt.ylabel("Total de personas infectadas")
plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
plt.show()
```

```
(array([-829399.50124244, 90006.63182773]), array([[ 3.27437846e+09, -2.726
28595e+08],
[-2.72628595e+08, 2.31133269e+07]]))
```



Contrarrestar con el modelo lineal

En base al modelo lineal y el modelo logístico he podido determinar que para los datos que yo tengo el modelo lineal representa un poco más el estado actual del coronavirus en Ecuador por qué este modelo se acopla a una recta en crecimiento mientras que el modelo logístico trata de demostrar que va a existir una baja de casos mientras que los datos reales me muestran un crecimiento exponencial.

Etapas del Desarrollo del Modelo

Para detallar mejor los pasos vamos a ir ejemplificando con los datos usados anteriormente para explicar el modelo logístico

- **Formulación del Problema.**

Determina el objeto de la simulación. En este caso COVID en Ecuador Se deben especificar claramente estos elementos:

- Resultados: Datos que esperamos tener aplicando modelos matematicos para entrenar al sistema |En esta caso modelo lineal o modelo logístico
- Plan de experimentación: Recopilacion de la informacion | En esta caso CSV con datos de positivos y recuperados
- Variables de interés: Las variables con las que se va a trabajar con el csv. | Del csv que tenemos se uso las columnas: created_at, positivas_pcr y hospitalizadas_altas.

- **Definición del sistema.** El sistema que se simulará debe estar definido perfectamente. Se debe establecer donde estará la frontera de interacción entre el sistema a estudiar y el medioambiente. | En este caso se hablamos de predecir los casos que vamos a tener

- **Formulación del modelo**

En esta etapa se capturan los aspectos relevantes del sistema real. Estos aspectos dependen de la formulación del problema.

Se analizara datos como las muestras tomadas a cada paciente, estas muestras son por cada paciente que llega con sintomas o personas que se realizan dicha prueba, tambien se tiene los datos probables que existen de muertes confirmadas, pruebas rezagadas, muestras pcr, etc

- **Colección de datos**

La naturaleza y la cantidad de datos se determinan por la formulación del problema y del modelo. | Estos datos fueron obtenidos desde un link de github:

https://raw.githubusercontent.com/andrab/ecuacovid/master/datos_crudos/ecuacovid.csv
(https://raw.githubusercontent.com/andrab/ecuacovid/master/datos_crudos/ecuacovid.csv)

- **Implementación del modelo en el ordenador**

Se implementa el modelo a través de un lenguaje de programación/simulación. | Para este ejemplo usamos Python y Jupyter

- **Verificación**

En esta etapa se verifica que no se hayan cometido errores durante la implementación del modelo. | Verificación por parte del estante

- **Diseño de experimentos** Se especifica las características de los experimentos a realizar:

- tiempo de arranque
- tiempo de simulación
- número de simulaciones.
- No se debe incluir la elaboración del conjunto de alternativas a probar para seleccionar la mejor, ya que esto es tarea de la optimización.

Para este caso se mostraron los experimentos en la regresión lineal

- **Experimentación**

En esta etapa se realizan simulaciones de acuerdo al diseño previo. Se recolectan y procesan los resultados.

- **Interpretación**

Se analiza la sensibilidad del modelo con respecto a los parámetros con mayor incertidumbre. Se podría decir que con los datos que estamos trabajando el covid va en creciente en Ecuador

- **Implementación**

Se entrega la solución al cliente y se lo capacita en su uso. Como se puede evidenciar, se ha ido aplicando los pasos para realizar un ejemplo de análisis de casos por covid, estos casos están analizados aplicando dos algoritmos el de regresión lineal, y el modelo logístico, para dicho análisis se ha tomado en cuenta un conjunto de datos que sirve para

- **Documentación** Elaboración de documentación técnica y manuales de uso. En el presente documento se encuentra detallado los pasos seguidos y como debe ir implementando los métodos para realizar los cálculos.

Todo este cuaderno

Conclusiones:

Con estos modelos matemáticos tanto el modelo lineal como el modelo logístico y con una base de datos completa y sobre todo que tenga datos reales podríamos trabajar en simulaciones para tratar de predecir que pasará en el futuro es decir tratar de ver cómo se mostrarán los datos en un futuro cercano o lejano de acuerdo a la cantidad de datos que tengamos en nuestro bd