

Universidad Politécnica Salesiana

Nombre: David Egas

Desarrollo

Aplicar el modelo logístico a el dataset actualizado de covid 19 en Ecuador, comparar el modelo logistico con el modelo lineal y dar opinión sobre cuál es el mejor modelo hasta ahora para la predicción de datos.

```
In [15]: 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
```

Carga del Dataset

```
In [16]: # Carga del dataset
dataset = pd.read_csv('datacovid.csv')
dataset
```

```
Out[16]:
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths
0	ABW	North America	Aruba	2020-03-13	2.0	2.0	NaN	
1	ABW	North America	Aruba	2020-03-19	NaN	NaN	0.286	
2	ABW	North America	Aruba	2020-03-20	4.0	2.0	0.286	
3	ABW	North America	Aruba	2020-03-21	NaN	NaN	0.286	
4	ABW	North America	Aruba	2020-03-22	NaN	NaN	0.286	
...
54385	NaN	NaN	International	2020-10-30	696.0	NaN	NaN	
54386	NaN	NaN	International	2020-10-31	696.0	NaN	NaN	
54387	NaN	NaN	International	2020-11-01	696.0	NaN	NaN	
54388	NaN	NaN	International	2020-11-02	696.0	NaN	NaN	
54389	NaN	NaN	International	2020-11-03	696.0	NaN	NaN	

54390 rows × 9 columns

Seleccionamos las columnas solo para Ecuador de la fecha y el total de casos

```
In [17]: dataset = dataset[dataset['location'].isin(['Ecuador'])] #Filtro la Informacion solo p
dataset = dataset.loc[:,['date','total_cases']] #Selecciono las columnas de analisis
# Expresar Las fechas en numero de dias desde el 01 Enero
FMT = '%Y-%m-%d'
date = dataset['date']
dataset['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime('
dataset
```

Out[17]:

	date	total_cases
14369	0	NaN
14370	1	NaN
14371	2	NaN
14372	3	NaN
14373	4	NaN
...
14673	304	166302.0
14674	305	167147.0
14675	306	168192.0
14676	307	169194.0
14677	308	169562.0

309 rows × 2 columns

Eliminamos las filas con valores nulos

```
In [18]: dataset2 = dataset.replace(-1, np.NaN).dropna()
dataset2
```

Out[18]:

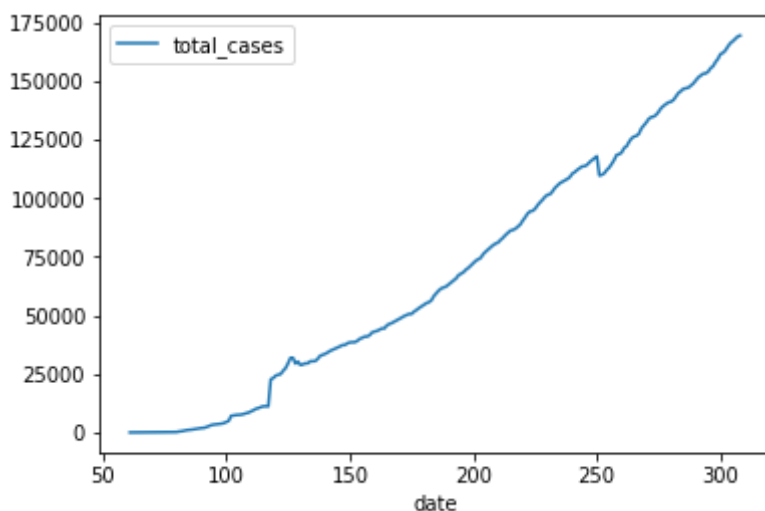
	date	total_cases
14430	61	1.0
14431	62	6.0
14432	63	7.0
14434	65	10.0
14435	66	13.0
...
14673	304	166302.0
14674	305	167147.0
14675	306	168192.0
14676	307	169194.0

	date	total_cases
14677	308	169562.0

243 rows × 2 columns

```
In [19]: dataset2.plot(x='date', y='total_cases')
```

```
Out[19]: <AxesSubplot:xlabel='date'>
```



Presentamos el dataset limpio de valores nulos

```
In [20]: dataset2
```

```
Out[20]:
```

	date	total_cases
14430	61	1.0
14431	62	6.0
14432	63	7.0
14434	65	10.0
14435	66	13.0
...
14673	304	166302.0
14674	305	167147.0
14675	306	168192.0
14676	307	169194.0
14677	308	169562.0

243 rows × 2 columns

Aplicando el modelo lineal

```
In [30]: x = list(dataset2.iloc[:, 0]) # Fecha
y = list(dataset2.iloc[:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
```

```
regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1), y)

# Veamos Los coeficienetes obtenidos, En nuestro caso, serán La Tangente
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
```

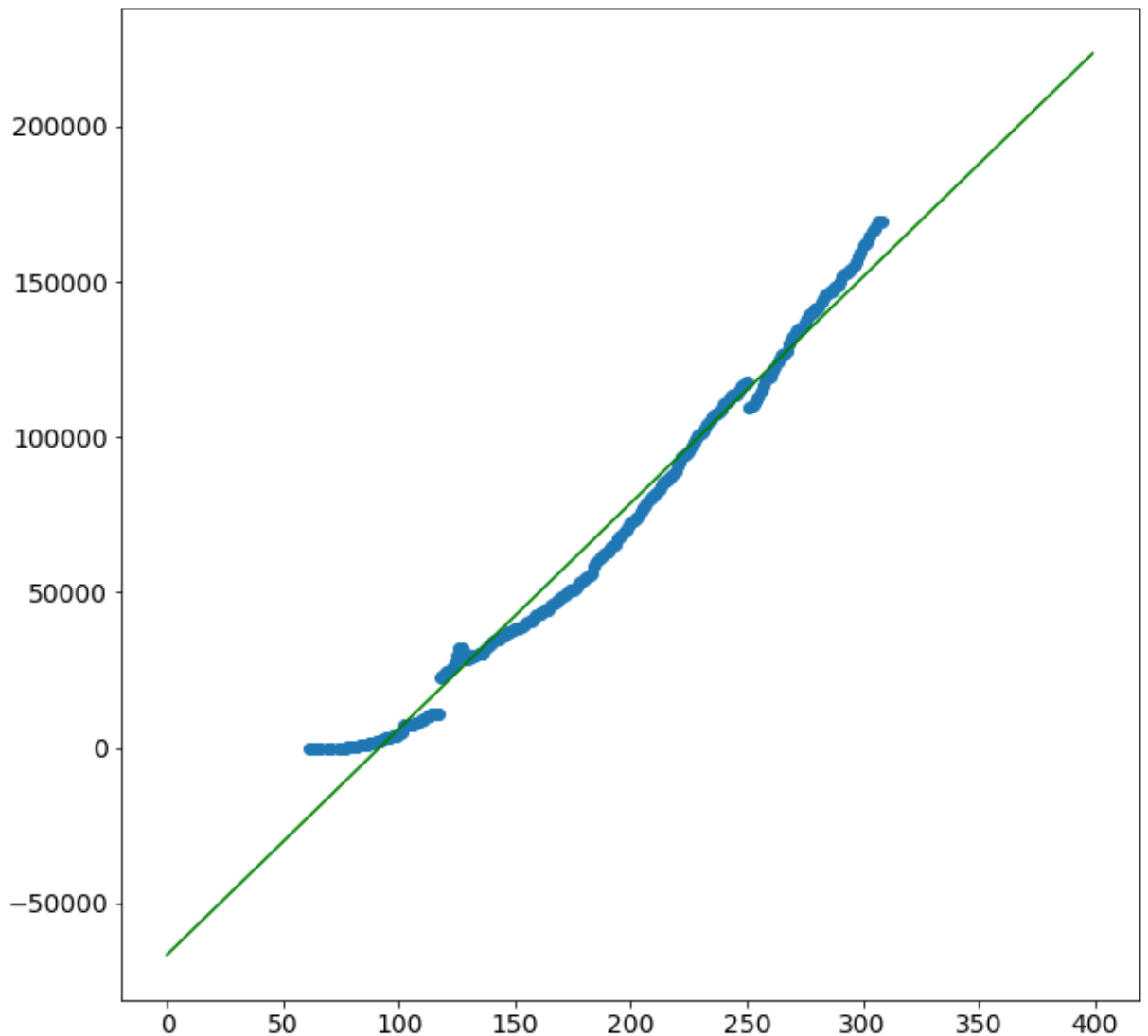
```
Coefficients:
[727.08534877]
Independent term:
-66583.68254486444
```

```
In [31]: y_prediccion = regr.predict([[315]])
print("El número de infectados el 10 de noviembre del 2020 será: ", int(y_prediccion))

y_prediccion = regr.predict([[336]])
print("El número de infectados el 1 de diciembre del 2020 será: ", int(y_prediccion))
```

```
El número de infectados el 10 de noviembre del 2020 será: 162448
El número de infectados el 1 de diciembre del 2020 será: 177716
```

```
In [32]: #Graficar
plt.scatter(x, y)
x_real = np.array(range(0, 400))
#print(x_real)
plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='green')
plt.show()
```



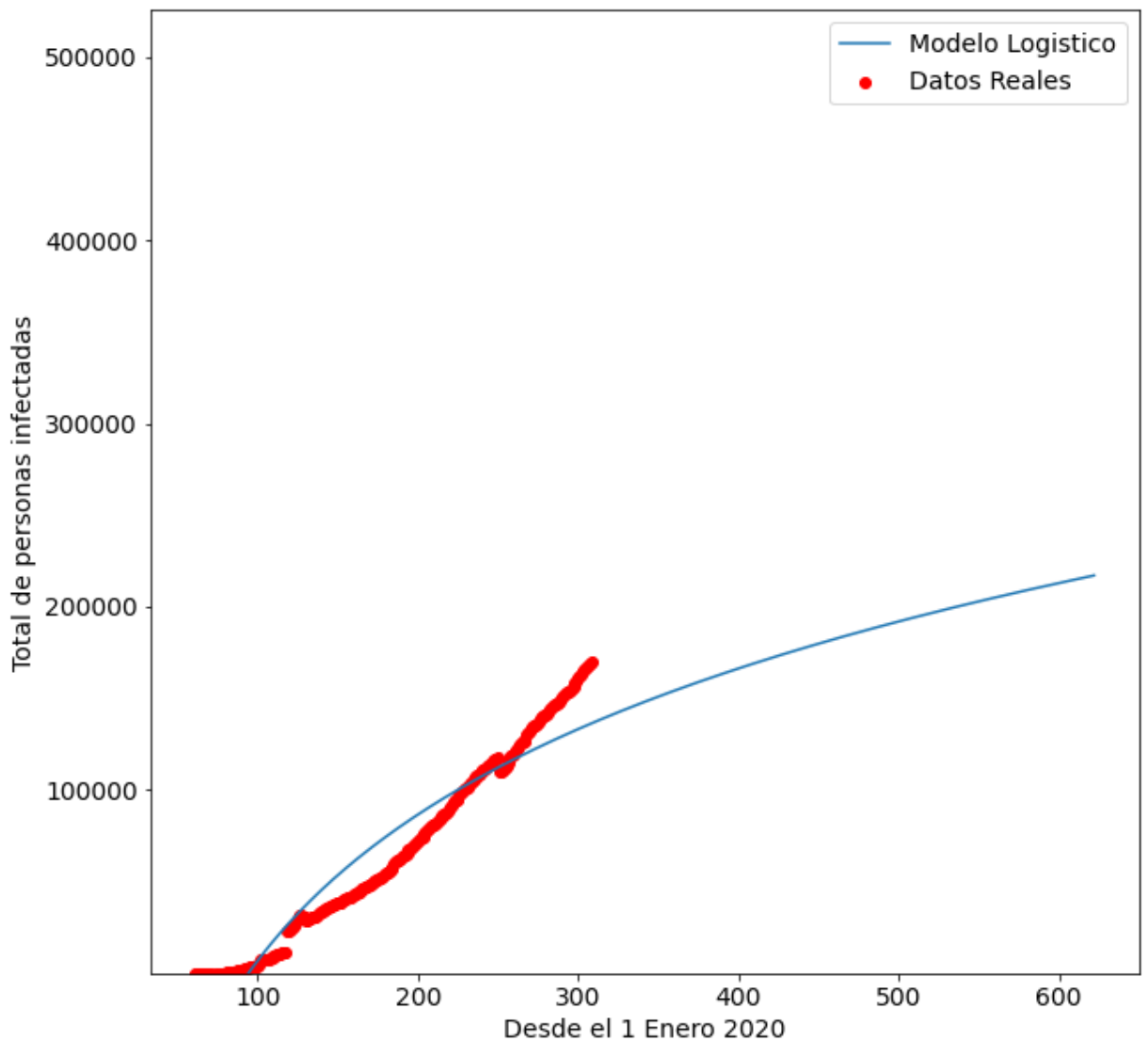
Aplicando el modelo Logistico

```
In [33]: 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([-522474.1438493 , 114959.20607545]), array([[ 1.58892449e+08, -3.06569296e+07],
          [-3.06569296e+07, 5.95553059e+06]]))

In [38]: pred_x = list(range(min(x),max(x)+315)) # Predecir 50 días mas
          plt.rcParams['figure.figsize'] = [10, 10]
          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="Predicted")
          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 límites de Y
          plt.show()
          print (exp_fit)
```



```
(array([-522474.1438493 , 114959.20607545]), array([[ 1.58892449e+08, -3.06569296e+07],
          [-3.06569296e+07, 5.95553059e+06]]))
```

Comparativa

El modelo Lineal para la fecha de 10 de noviembre del 2020 predice 162448 infectados, mientras que el modelo logístico predice 140000 infectados mas o menos, según los datos reales, el que mas se acerca a la realidad es el modelo lineal, y para mí, este es el mejor modelo para predecir hasta ahora.

Definición del sistema

El sistema se implementará como una herramienta a tener presente ahora que nuestro país esta afrontando la crisis del coronavirus, estos modelo debe ser usado como referencia sobre el crecimiento de los casos y así tener una idea clara de las fechas en las que la situación empeorará o mejorará.

Formulación del modelo

Este sistema usa datos históricos para predecir el avance en número de infectados por coronavirus en el país, se consideran dos entradas, las fechas y el número total de casos, luego se realiza un entrenamiento de regresión lineal y logistica y generamos un posible número de infectados a futuro.

Colección de datos

Los datos que vamos a usar contienen información variada como: fechas, numero de casos, numero de muertes, muertes dependiendo los factores médicos, etc. Obtenidos de datos mundiales por la OMS.

Implementación, Verificación y Diseño de experimentos

Esta parte estará implementada en la parte de arriba del cuaderno de Jupyter, donde realizamos el entrenamiento y algunos experimentos como por ejemplo el número de posibles contagiados hasta el 10 de noviembre de 2020.

In []: