

# UNIVERSIDAD POLITÉCNICA SALESIANA

Nombre: David Egas

## Desarrollo

Realizar un modelo de regresión lineal para determinar los futuros casos de Coronavirus en Ecuador, tomando en cuenta un dataset que contienen los datos históricos de contagiados en diferentes partes del mundo incluido Ecuador, y las fechas desde el día 0 en el país hasta el mes de noviembre de 2020.

```
In [47]: # 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
```

## Infectados

Dataset utilizado: este dataset esta tomando de la página web Ourl World in Data.

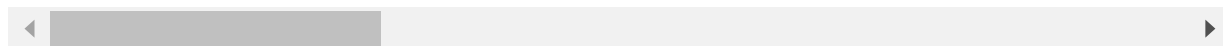
```
In [48]: # Carga del dataset
dataset = pd.read_csv('datacovid.csv')
dataset
```

```
Out[48]:
```

	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	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_d
<b>54388</b>	NaN	NaN	International	2020-11-02	696.0	NaN	NaN	
<b>54389</b>	NaN	NaN	International	2020-11-03	696.0	NaN	NaN	

54390 rows × 49 columns



## Seleccionamiento:

Se toman los valores solo para Ecuador, desde la primera fecha hasta la última y el número de casos encontrados.

```
In [49]: 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[49]:

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

309 rows × 2 columns

## Limpieza de los datos

En los primeros días no se tenían casos por lo que nos dará un valor de NaN, esto puede causar problemas en el entrenamiento por lo que vamos a reemplazar por 0.

```
In [50]: dataset = dataset.fillna(0)
dataset
```

Out[50]:

	date	total_cases
<b>14369</b>	0	0.0

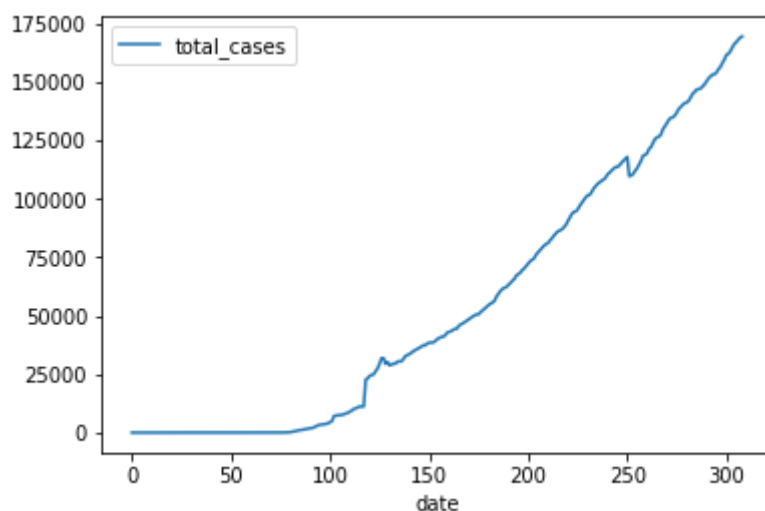
	date	total_cases
<b>14370</b>	1	0.0
<b>14371</b>	2	0.0
<b>14372</b>	3	0.0
<b>14373</b>	4	0.0
...	...	...
<b>14673</b>	304	166302.0
<b>14674</b>	305	167147.0
<b>14675</b>	306	168192.0
<b>14676</b>	307	169194.0
<b>14677</b>	308	169562.0

309 rows × 2 columns

## Gráfica de el número de casos respecto a las fechas

```
In [51]: dataset.plot(x='date', y='total_cases')
```

```
Out[51]: <AxesSubplot:xlabel='date'>
```



## Entrenamiento de Regresión Lineal

```
In [52]: x = list(dataset.iloc[:, 0]) # Fecha
y = list(dataset.iloc[:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()
```

```
In [54]: # Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1), y)
```

```
Out[54]: LinearRegression()
```

```
In [55]: # Veamos Los coeficientes 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:
[580.86758941]
Independent term:
-34959.87090510494
```

## Probando el modelo

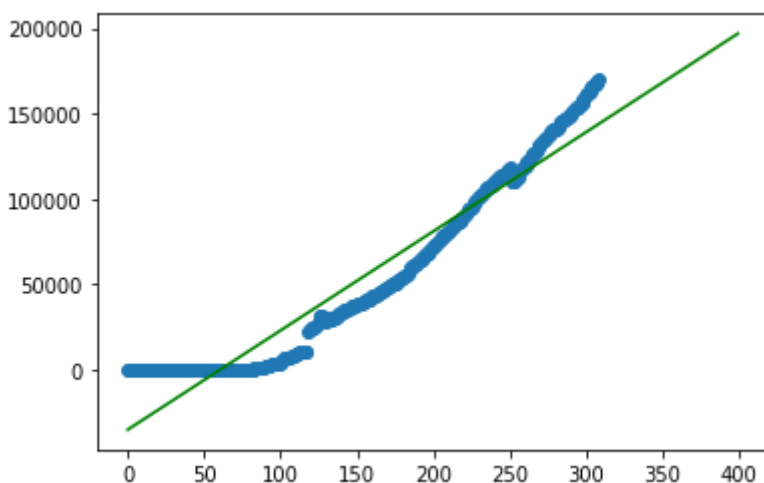
Predeciremos cuantos casos de covid habrán el día 365 y veremos si el resultado tiene lógica frente a los obtenidos ya anteriormente

```
In [61]: 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: 148013
El número de infectados el 1 de diciembre del 2020 será: 160211
```

```
In [62]: #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()
```



## Curados

Dataset utilizado: este dataset esta tomando de Johns Hopkins University Center for Systems Science and Engineering

```
In [66]: # Carga del dataset
dataset = pd.read_csv('datasetcovid2.csv')
dataset
```

```
Out[66]:
```

	Date	Country	Confirmed	Recovered	Deaths
0	2020-01-22	Afghanistan	0	0	0
1	2020-01-23	Afghanistan	0	0	0
2	2020-01-24	Afghanistan	0	0	0
3	2020-01-25	Afghanistan	0	0	0

	Date	Country	Confirmed	Recovered	Deaths
4	2020-01-26	Afghanistan	0	0	0
...	...	...	...	...	...
54145	2020-10-28	Zimbabwe	8320	7845	242
54146	2020-10-29	Zimbabwe	8349	7864	242
54147	2020-10-30	Zimbabwe	8362	7884	242
54148	2020-10-31	Zimbabwe	8367	7894	243
54149	2020-11-01	Zimbabwe	8374	7927	243

54150 rows × 5 columns

## Seleccionamiento:

En este caso se toma los datos de las fechas y el número de recuperados de coronavirus en Ecuador desde la fecha 0 hasta el presente mes.

```
In [67]: dataset = dataset[dataset['Country'].isin(['Ecuador'])] #Filtro la Informacion solo por Ecuador
dataset = dataset.loc[:,['Date','Recovered']] #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('2020-01-01', FMT)).days)
dataset
```

```
Out[67]:
```

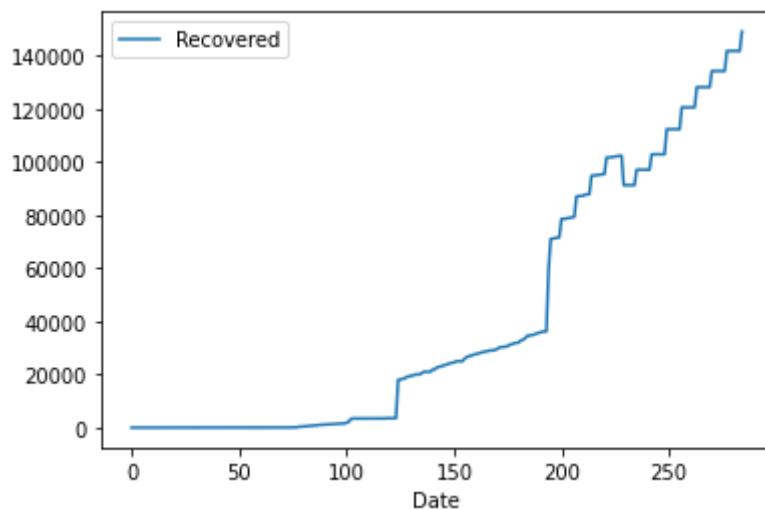
	Date	Recovered
14820	0	0
14821	1	0
14822	2	0
14823	3	0
14824	4	0
...	...	...
15100	280	141759
15101	281	141759
15102	282	141759
15103	283	141759
15104	284	149048

285 rows × 2 columns

## Gráfica de los datos

```
In [68]: dataset.plot(x='Date', y='Recovered')
```

```
Out[68]: <AxesSubplot:xlabel='Date'>
```



## Entrenamiento y aplicación de regresión lineal

```
In [69]: x = list(dataset.iloc[:, 0]) # Fecha
y = list(dataset.iloc[:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()
```

```
In [71]: # Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1), y)
```

Out[71]: LinearRegression()

```
In [72]: # Veamos Los coeficientes 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
```

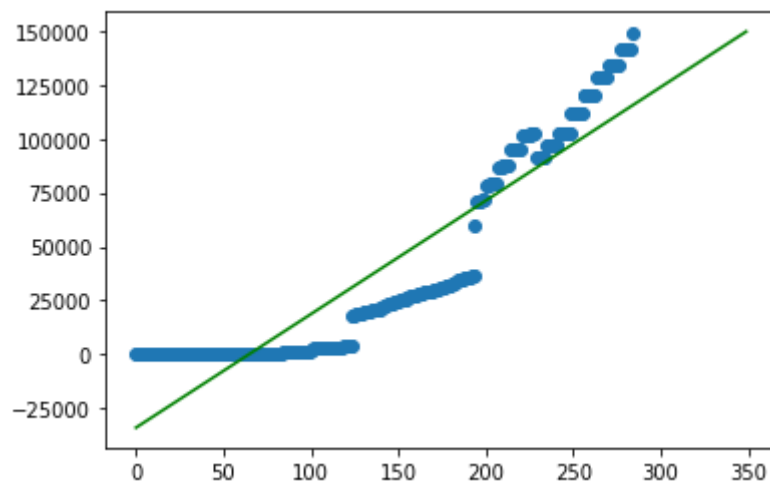
```
Coefficients:
[526.82761331]
Independent term:
-34097.67196662989
```

```
In [74]: y_prediccion = regr.predict([[293]])
print("El número de recuperados a la fecha de 10 de noviembre de 2020 será:", int(y_pr

y_prediccion = regr.predict([[314]])
print("El número de recuperados a la fecha de 1 de diciembre de 2020 será: ", int(y_pr
```

```
El número de recuperados a la fecha de 10 de noviembre de 2020 será: 120262
El número de recuperados a la fecha de 1 de diciembre de 2020 será: 131326
```

```
In [76]: #Graficar
plt.scatter(x, y)
x_real = np.array(range(0, 350))
#print(x_real)
plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='green')
plt.show()
```



## Resultados:

La comparación nos deja que:

- Para el 1 de Diciembre habrá 160211 personas infectadas de covid 19 en Ecuador
- Al mismo tiempo para el 1 de Diciembre habrá 131326 personas recuperadas de covid 19 en Ecuador

Los datos estan basados en históricos hasta los primeros días de noviembre.

In [ ]: