

MODELO LINEAL

AUTOR: CARLOS MOROCHO

```
In [1]: # Importamos las librerías necesarias
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
```

```
In [23]: # Cargamos el dataset
datos = pd.read_csv('DatosCOVID2020.csv', sep = ',')
datos
```

Out[23]:

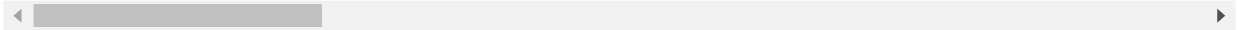
	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muerteres_confirmadas	m
0	129	129	0	106	1	
1	206	206	77	178	2	
2	273	273	67	236	2	
3	354	354	81	296	2	
4	762	762	408	651	2	
5	762	762	0	282	3	
6	1153	1153	391	481	4	
7	1670	1670	517	711	7	

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muerteres_confirmadas	m
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	
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	

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muertes_confirmadas	m
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	
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	

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muertes_confirmadas	m
244	605331	584759	4590	50256	8614	
245	610265	589693	4934	50604	8642	
246	614531	593959	4266	50485	8658	

247 rows × 32 columns



1) Predicción de nuevos casos positivos (Próxima semana, Próximo mes)

```
In [24]: #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[24]:

	positivas	created_at
0	23	72
1	28	73
2	37	74
3	58	75
4	111	76
5	168	77
6	260	78

6	200	78
	positivas	created_at
7	426	79
8	532	80
9	789	81
10	981	82
11	1082	83
12	1211	84
13	1403	85
14	1627	86
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
...
317	160615	200

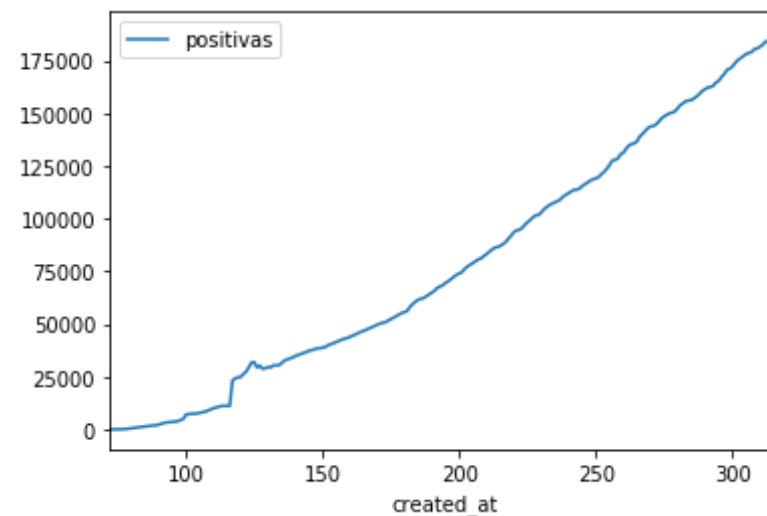
217	160015	289
	positivas	created_at
218	161378	290
219	162245	291
220	162379	292
221	163071	293
222	164581	294
223	165407	295
224	167226	296
225	168570	297
226	170591	298
227	171134	299
228	172148	300
229	173864	301
230	175258	302
231	176103	303
232	177148	304
233	178150	305
234	178518	306
235	179066	307
236	180389	308
237	180739	309
238	181464	310
239	182442	311
240	183863	312
241	184225	313
242	184667	314

242	184007	314
	positivas	created_at
243	185586	315
244	186469	316
245	187630	317
246	188583	318

247 rows × 2 columns

```
In [25]: # Graficamos la cantidad de infectados
datos_infectados.plot(x = 'created_at', y = 'positivas')
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x25258adc438>



```
In [27]: #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

#Creación y entranamiento del modelo
modelo_lineal = linear_model.LinearRegression()
```

```
modelo_lineal.fit(np.array(x).reshape(-1, 1), y)

# Veamos los coeficientes obtenidos, En nuestro caso, serán la Tangente
e
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
```

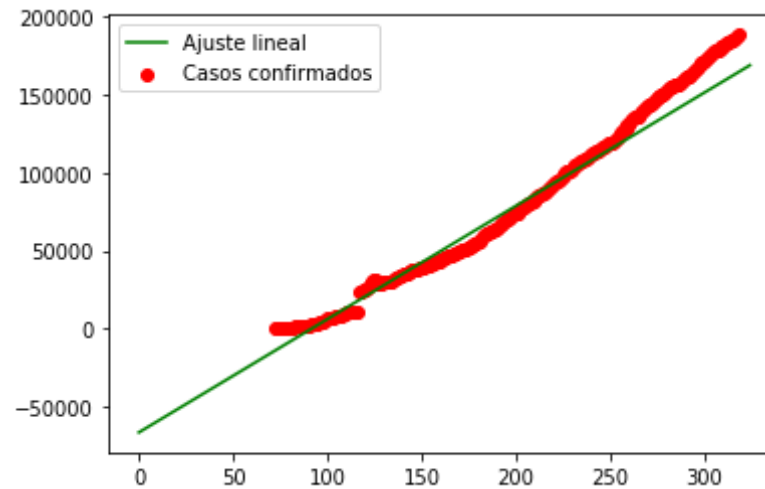
```
Coefficients:
[727.08534877]
Independent term:
-66583.68254486444
```

```
In [28]: #Predicción para una semana
y_prediccion = regr.predict([[315]])
print("El número de infectados el 10 de noviembre del 2020 será: ", int
(y_prediccion))

#Gráfica de los datos y la predicción para una semana
x_real = np.array(range(0, 325))

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

El número de infectados el 10 de noviembre del 2020 será: 162448

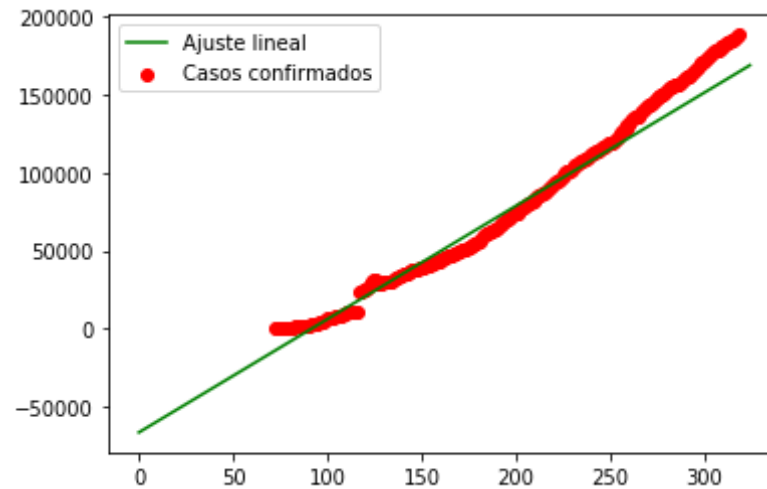


```
In [29]: #Predicción para una semana
y_prediccion = regr.predict([[336]])
print("El número de infectados el 1 de diciembre del 2020 será: ", int(
y_prediccion))

#Gráfica de los datos y la predicción para una semana
x_real = np.array(range(0, 325))

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

El número de infectados el 1 de diciembre del 2020 será: 177716



2) Predicción de casos con pacientes recuperados (Próxima semana, Próximo mes)

```
In [30]: #Se filtran los datos para obtener solo los casos recuperados
datos_recuperados = datos.loc[:, ['hospitalizadas_altas', 'created_at']]

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

datos_recuperados
```

Out[30]:

	hospitalizadas_altas	created_at
0	0	72
1	0	72

	hospitalizadas_altas	created_at
2	0	74
3	0	75
4	0	76
5	0	77
6	3	78
7	3	79
8	3	80
9	3	81
10	3	82
11	3	83
12	3	84
13	3	85
14	3	86
15	3	87
16	3	88
17	54	89
18	58	90
19	58	91
20	65	92
21	71	93
22	100	94
23	100	95
24	100	96
25	140	97

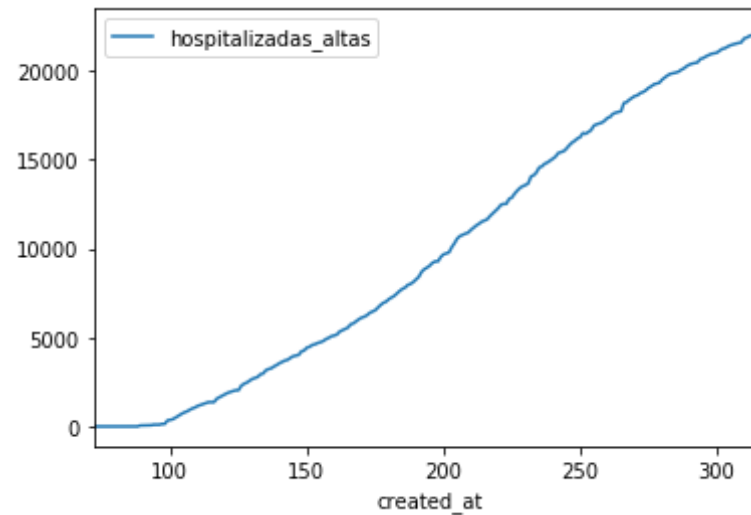
26	140	98
hospitalizadas_altas	created_at	
27	339	99
28	368	100
29	411	101
...
217	20249	289
218	20338	290
219	20394	291
220	20432	292
221	20464	293
222	20656	294
223	20700	295
224	20794	296
225	20869	297
226	20940	298
227	20969	299
228	21004	300
229	21112	301
230	21219	302
231	21260	303
232	21349	304
233	21423	305
234	21476	306
235	21516	307
236	21552	308

237	21589	309
	hospitalizadas_altas	created_at
238	21806	310
239	21857	311
240	21946	312
241	21994	313
242	22028	314
243	22099	315
244	22195	316
245	22242	317
246	22346	318

247 rows × 2 columns

```
In [31]: # Graficamos los pacientes recuperados
datos_recuperados.plot(x = 'created_at', y = 'hospitalizadas_altas')
```

```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x25258b55ef0>
```



```
In [32]: #Obtención de las variables para el entrenamiento
x = list(datos_recuperados.iloc[:, 1]) #Fecha (Número de día)
y = list(datos_recuperados.iloc[:, 0]) #Numero de recuperados

#Creación y entranamiento del modelo
modelo_lineal = linear_model.LinearRegression()
modelo_lineal.fit(np.array(x).reshape(-1, 1), y)

# Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangent
e
print('Coeficientes: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
```

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

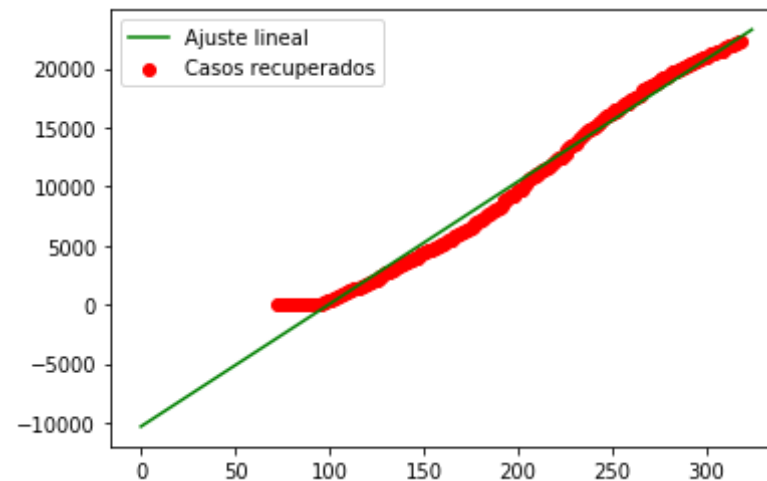
```
In [37]: #Predicción para una semana
y_prediccion = modelo_lineal.predict([[315]])
print("El número de infectados el 10 de noviembre del 2020 será: ", int
```

```
(y_prediccion))

#Gráfica de los datos y la predicción para una semana
x_real = np.array(range(0, 325))

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

El número de infectados el 10 de noviembre del 2020 será: 22366



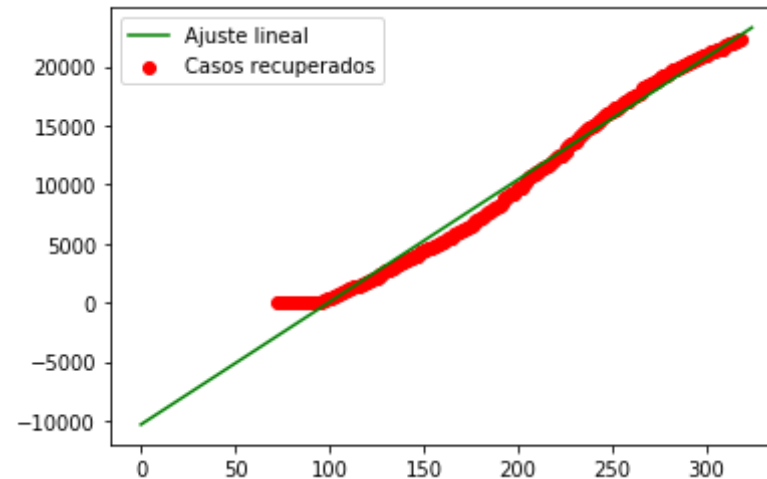
```
In [38]: #Predicción para un mes
y_prediccion = modelo_lineal.predict([[336]])
print("El número de infectados el 1 de diciembre del 2020 será: ", int(
y_prediccion))

#Gráfica de los datos y la predicción para un mes
x_real = np.array(range(0, 325))

plt.scatter(x, y , color = 'red')
plt.plot(x_real, modelo_lineal.predict(x_real.reshape(-1, 1)), color =
'green')
```

```
plt.legend(('Ajuste lineal', 'Casos recuperados'))  
plt.show()
```

El número de infectados el 1 de diciembre del 2020 será: 24546



Analisis

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.

Conclusion

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 logística y generamos un posible número de infectados a futuro.

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

Los casos sin duda han en aumentos, pero aunque esto parezca desalentador existe un punto que estamos a punto de tocar, el de máximo de infectados, actualmente las medidas en todo el mundo hacen que la gente inconscientemente se proteja, claro no es suficiente pero al menos contrastará la gran tasa de infectados que seguiremos viendo con el pasar de los días hasta que no llegue una vacuna.

In []: