

Wilson Rodas

Modelo Lineal

```
In [9]: #Importación de las librerías necesarias
import pandas as pd
import numpy as np
from datetime import datetime
from sklearn import linear_model
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [17]: #Obtención de los datos de COVID-19 de Ecuador
url_datos = 'DatosCOVID2020.csv'

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

```
Out[17]:
```

	muestras	muestras_pcr	muestras_pcr_nuevas	pruebas_rezagadas	muerteres_confirmadas	muerteres_probables	muerteres	muerteres_nuevas	positivas
0	129	129	0	106	1	0	1	0	23
1	206	206	77	178	2	0	2	1	28
2	273	273	67	236	2	0	2	0	37
3	354	354	81	296	2	0	2	0	58
4	762	762	408	651	2	0	2	0	111
...
242	597099	576527	559	49598	8525	4324	12849	10	184667
243	600741	580169	3642	49067	8592	4328	12920	71	185586
244	605331	584759	4590	50256	8614	4332	12946	26	186469
245	610265	589693	4934	50604	8642	4335	12977	31	187630
246	614531	593959	4266	50485	8658	4339	12997	20	188583

247 rows × 32 columns

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

```
In [21]: #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', formato)).days
)

datos_infectados
```

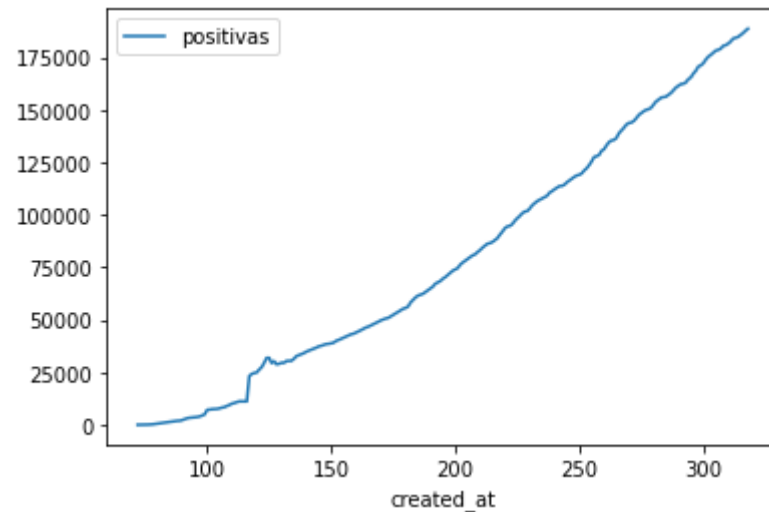
```
Out[21]:
```

	positivas	created_at
0	23	72
1	28	73
2	37	74
3	58	75
4	111	76
...
242	184667	314
243	185586	315
244	186469	316
245	187630	317
246	188583	318

247 rows × 2 columns

```
In [23]: datos_infectados.plot(x = 'created_at', y = 'positivas')
```

```
Out[23]: <AxesSubplot:xlabel='created_at'>
```



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 [24]: #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)
```

Out[24]: LinearRegression()

1.1 Predicción para una semana

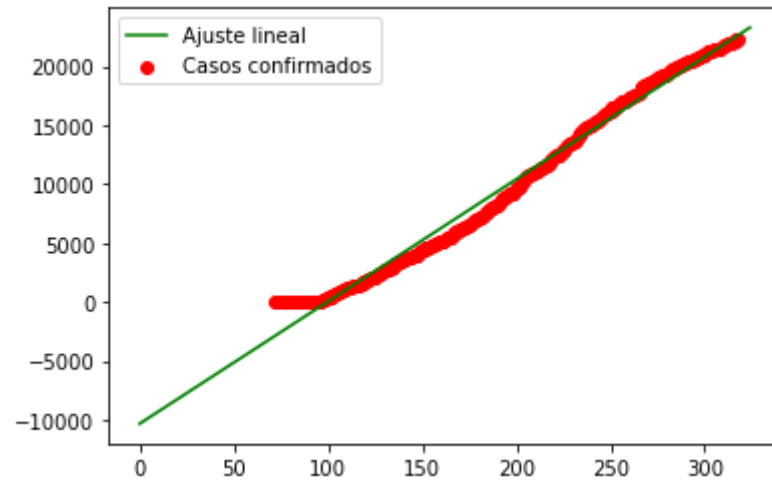
```
In [37]: #Comprobación de la predicción para 7 días despues del 14/11/2020
dia = x[-1] + 7 #Se obtiene el dia a predecir sumando la ultima fecha registrada mas una semana
prediccion_semana = modelo_lineal.predict([[dia]])
print('Predicción de casos positivos para el 21/11/2020: ', int(prediccion_semana))
```

Predicción de casos positivos para el 21/11/2020: 183410

```
In [54]: #Gráfica de los datos y la predicción
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 confirmados'))
plt.show()
```



De acuerdo al modelo entrenado, el número de casos que habrá el 21/11/2020 será 183410 confirmados.

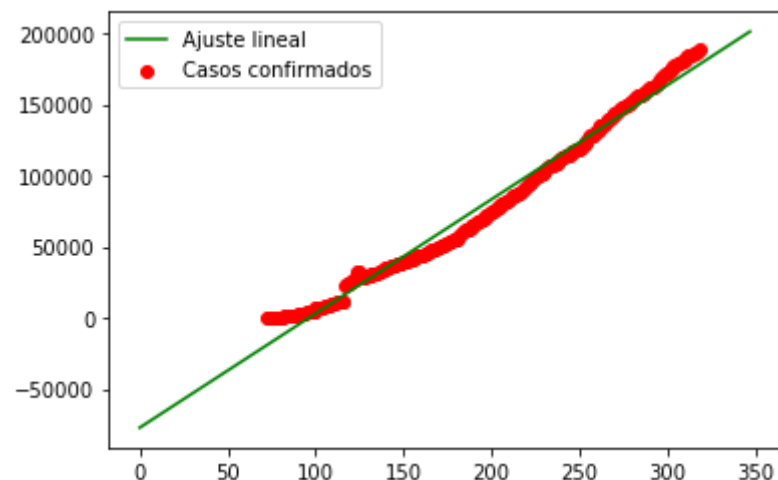
1.2 Predicción para un mes

```
In [34]: #Comprobación de la predicción para 30 días después del 14/11/2020
dia = x[-1] + 30 #Se obtiene el día a predecir sumando la última fecha registrada mas una semana
prediccion_mes = modelo_lineal.predict([[dia]])
print('Predicción de casos positivos para el 13/12/2020: ', int(prediccion_mes))
```

Predicción de casos positivos para el 13/12/2020: 201854

```
In [42]: #Gráfica de los datos y la predicción
x_real = np.array(range(0, 348))

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 confirmados'))
plt.show()
```



De acuerdo al modelo entrenado, el número de casos que habrá el 13/12/2020 será 201854 confirmados.

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

```
In [43]: #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', formato)).days
)

datos_recuperados
```

```
Out[43]:
```

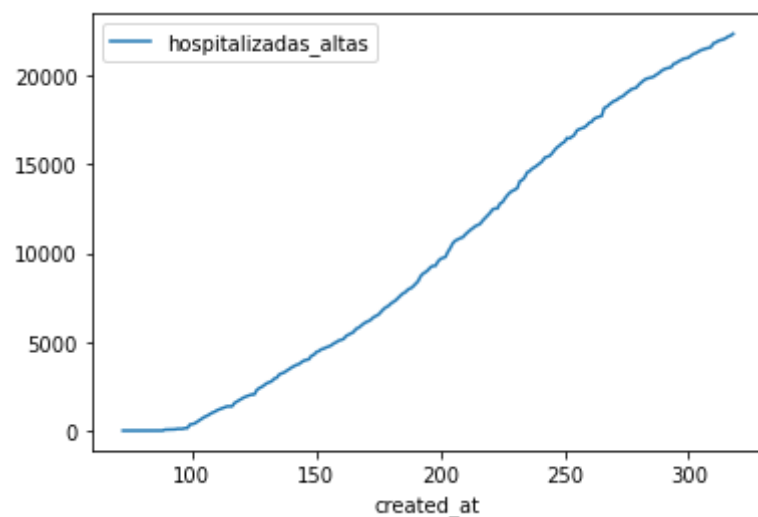
	hospitalizadas_altas	created_at
0	0	72
1	0	73
2	0	74
3	0	75
4	0	76
...

	hospitalizadas_altas	created_at
242	22028	314
243	22099	315
244	22195	316
245	22242	317
246	22346	318

247 rows × 2 columns

```
In [45]: datos_recuperados.plot(x = 'created_at', y = 'hospitalizadas_altas')
```

```
Out[45]: <AxesSubplot:xlabel='created_at'>
```



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 22346 casos recuperados.

```
In [48]: #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)
```

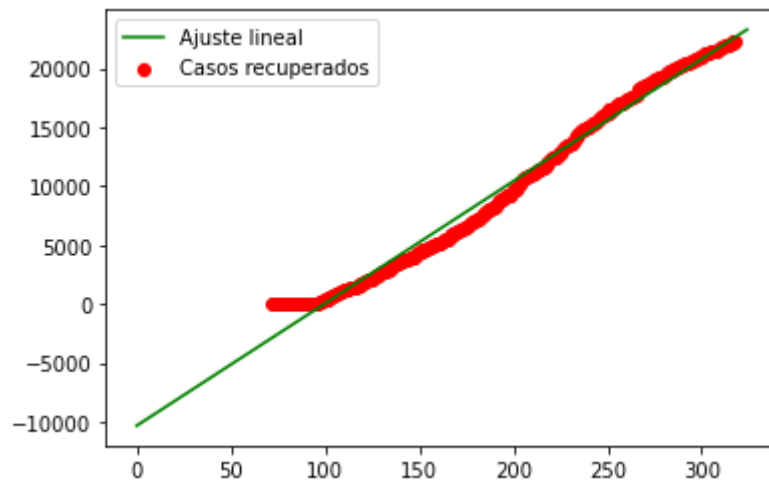
Out[48]: LinearRegression()

2.1 Predicción para una semana

```
In [51]: #Comprobación de la predicción para 7 días despues del 14/11/2020  
dia = x[-1] + 7 #Se obtiene el dia a predecir sumando la ultima fecha registrada mas una semana  
prediccion_semana = modelo_lineal.predict([[dia]])  
print('Predicción de casos recuperados para el 21/11/2020: ', int(prediccion_semana))
```

Predicción de casos recuperados para el 21/11/2020: 23404

```
In [50]: #Gráfica de los datos y la predicción  
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()
```



De acuerdo al modelo entrenado, el número de casos que habrá el 13/12/2020 será 23404 recuperados.

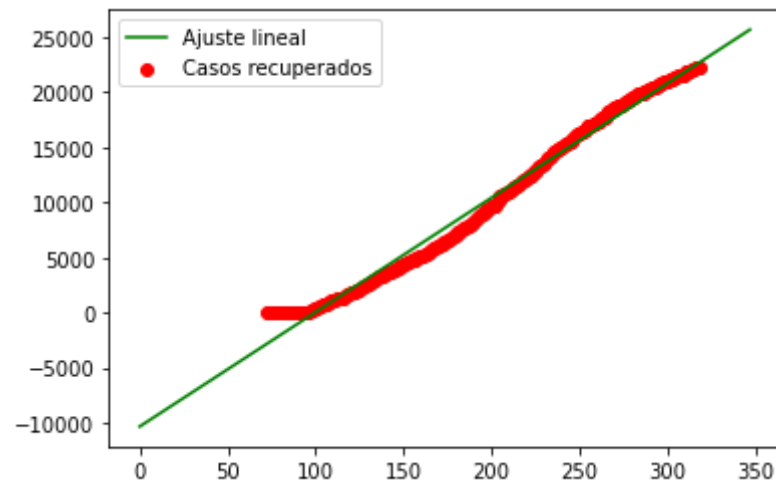
2.2 Predicción para un mes

```
In [52]: #Comprobación de la predicción para 30 días después del 14/11/2020
dia = x[-1] + 30 #Se obtiene el día a predecir sumando la última fecha registrada más una semana
prediccion_mes = modelo_lineal.predict([[dia]])
print('Predicción de casos recuperados para el 13/12/2020: ', int(prediccion_mes))
```

Predicción de casos recuperados para el 13/12/2020: 25792

```
In [53]: #Gráfica de los datos y la predicción
x_real = np.array(range(0, 348))

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()
```



De acuerdo al modelo entrenado, el número de casos que habrá el 13/12/2020 será 25792 recuperados.

3) Análisis

Los datos fueron obtenidos del siguiente enlace: <https://github.com/andrab/ecuacovid>, el cual es un proyecto que reúne los datos de los informes nacionales. Con respecto al modelo lineal entrenado en este trabajo, se ha utilizado un total de 246 registros, de los cuales los datos principales a tratar fueron: la fecha ('created_at'), los casos confirmados ('positivas'), y los casos recuperados ('hospitalizadas_altas').

4) Conclusiones

Ya que los datos presentados se mantienen bastante constantes y no varían demasiado en cuanto a linealidad, entonces se puede declarar que el modelo de regresión lineal podría ser la alternativa más acertada para predecir casos de COVID-19 durante la pandemia en Ecuador.

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

La emergencia sanitaria por COVID-19 en el Ecuador y en el resto del mundo es una situación realmente dura, siendo así que la gran mayoría de sistemas de salud se han visto saturados con el incremento y llegada de nuevos pacientes. La prioridad de las naciones debería ser apoyar a la investigación de la cura contra el virus, o caso contrario, apoyar a la compra de insumos médicos para respaldar a las casas de salud.