Modelo probabilístico

Nombre: David Egas

Desarrollo

Implementacion de un modelo probabilistico de infección por el virus Covid-19

Se realiza un análisis probabilistico simple del crecimiento de la infección en Python y el modelos para comprender mejor la evolución de la infección.

Se crea modelos de series temporales del número total de personas infectadas hasta la fecha (es decir, las personas realmente infectadas más las personas que han sido infectadas). Estos modelos tienen parámetros , que se estimarán por ajuste de probabilidad.

In [73]:

```
# Importar las librerias para el analasis
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 [75]:

```
# Carga del dataset
df = pd.read_csv('datacovid.csv')
df
```

Out[75]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothe
0	ABW	North America	Aruba	2020-03-13	2.0	2.0	Nal
1	ABW	North America	Aruba	2020-03-19	NaN	NaN	0.28
2	ABW	North America	Aruba	2020-03-20	4.0	2.0	0.28
3	ABW	North America	Aruba	2020-03-21	NaN	NaN	0.28
4	ABW	North America	Aruba	2020-03-22	NaN	NaN	0.28
...	...	...	...	...	...	...	.
54385	NaN	NaN	International	2020-10-30	696.0	NaN	Nal
54386	NaN	NaN	International	2020-10-31	696.0	NaN	Nal
54387	NaN	NaN	International	2020-11-01	696.0	NaN	Nal
54388	NaN	NaN	International	2020-11-02	696.0	NaN	Nal
54389	NaN	NaN	International	2020-11-03	696.0	NaN	Nal

54390 rows × 49 columns

Imprimos los resultados y agregamos el numero del dia

In [76]:

```
df = df[df['location'].isin(['Ecuador'])] #Filtro la Informacion solo para Ecuador
df = df[(df.total_cases >= 1)]
df = df.loc[:,['date', 'new_cases', 'total_cases']] #Selecciono las columnas de anal
asis
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("202
0-01-01", FMT)).days)
df
```

Out[76]:

	date	new_cases	total_cases
14430	60	1.0	1.0
14431	61	5.0	6.0
14432	62	1.0	7.0
14434	64	3.0	10.0
14435	65	3.0	13.0
...	...	...	...
14673	303	1394.0	166302.0
14674	304	845.0	167147.0
14675	305	1045.0	168192.0
14676	306	1002.0	169194.0
14677	307	368.0	169562.0

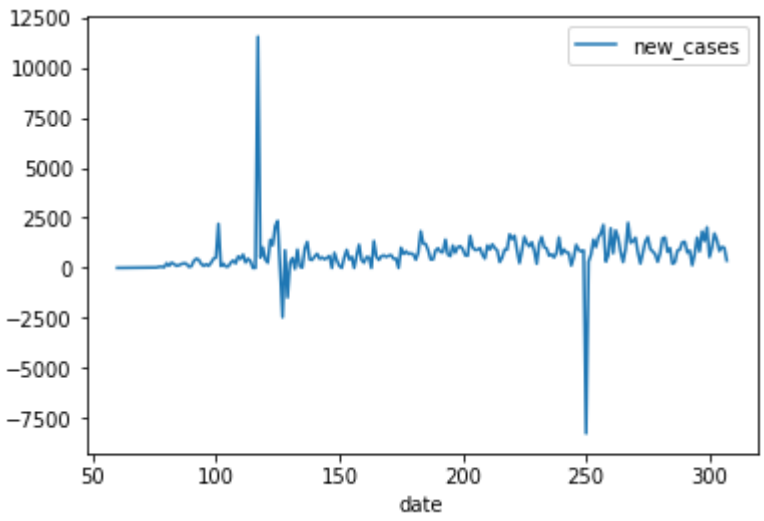
243 rows × 3 columns

In [77]:

```
df.plot(x ='date', y='new_cases')
```

Out[77]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a52231c408>
```



Ahora podemos analizar un modelo probabilisto para el examen.

El modelo basado en probabilidad

Para realizar un estimacion del factor de crecimiento de los casos de Covid 19 en Ecuador calculamos la mediana, con esto obtenemo el valor medio de crecimiento de un conjunto de datos, con esto podemos obtener un factor de crecimiento o taza de crecimiento de los nuevos casos.

In [78]:

```
filtro = df["new_cases"] # Filtro los datos que se empezo a tener casos
#Obtenemos la mediana
media = filtro.mean()
mediana = filtro.median()
print(mediana)
print(media)
```

```
660.0
697.7860082304527
```

De la ecuación de la recta y = mX + b nuestra pendiente «m» es el coeficiente y el término independiente «b»

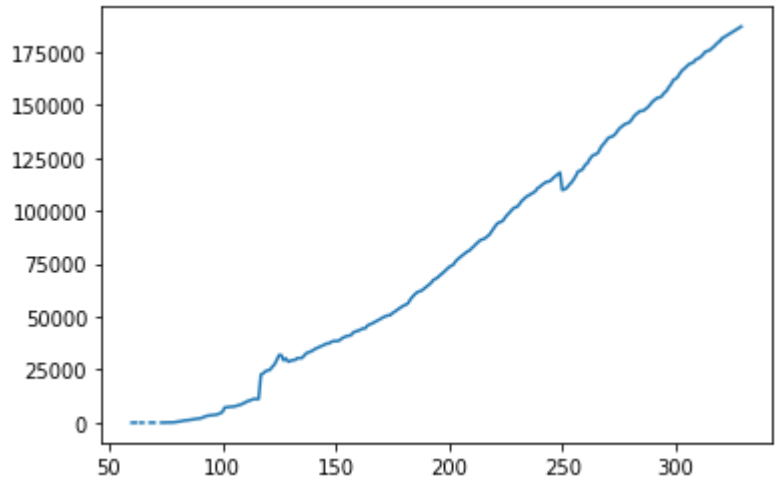
In [79]:

```
#Vamos a comprobar:
# según la media y la mediana podemos obtener la taza de crecieminto y predicir su c
omportamiento.
# Cargamos los datos de total de casos
url = 'https://covid.ourworldindata.org/data/ecdc/total_cases.csv'
df_t = pd.read_csv(url)
FMT = '%Y-%m-%d'
date = df_t['date']
df_t['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2
020-01-01", FMT)).days)
df_t = df_t.loc[:,['date','Ecuador']] #Selecciono las columnas de analasis
y = list(df_t.iloc [:, 1]) # Total casos
x = list(df_t.iloc [:, 0]) # Dias
#Realizamos un ejemplo de prediccion
prediccion_siguiente = int(y[-1] + mediana)
print(prediccion_siguiente)
```

```
180955
```

In [80]:

```
# Quiero predecir cuántos "Casos" voy a obtener de aqui a 10 dias.
for i in range(x[-1], x[-1]+10):
    x.append(i)
    y.append(int(y[-1] + mediana))
plt.plot(x[61:], y[61:])
plt.show()
```



Practica

1. Comparar el modelo de predicion matematico vs probabilidad.
2. Retroceder un semana y comparar el modelo matematico vs probabilidad vs reales. Solo cargan los datos para generar los modelos menos 7 dias.

Puntos extras: Investigas sobre la correlacion de variables y aplicar el calculo en base a los datos del Ecuador.

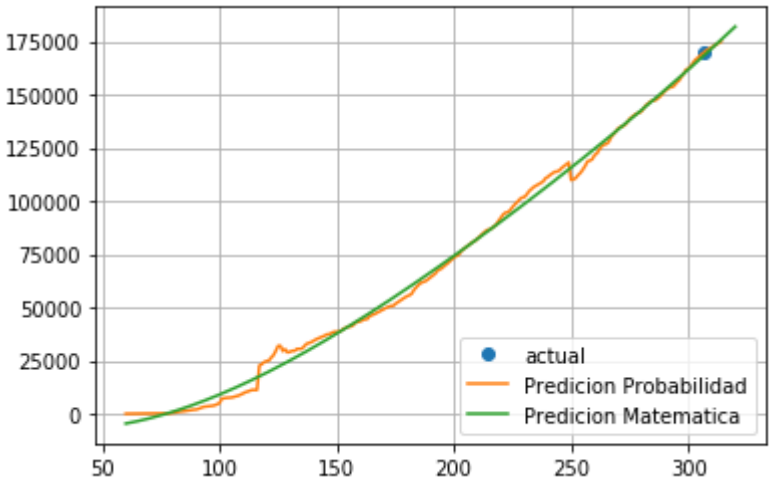
1. Comparar el modelo de predicion matematico vs probabilidad.

In [81]:

```
#Implementar
# predicion probabilidad
x = list(df.iloc [:, 0]) # Dias
y = list(df.iloc [:, 2]) # Total casos
#Realizamos un ejemplo de prediccion
prediccion_siguiente = int(y[-1] + mediana)

plt.plot(x[-1],y[-1],'o',label='actual')

#predicion a una semana
for i in range(x[-1], x[-1]+8):
    x.append(i)
    y.append(int(y[-1] + mediana))

print(y)
#predicion matematica
x1 = np.array(x)
y1 = np.array(y)
def func_polinomial(x, a, b, c, d):
    return a*x**4 + b*x**3 + c*x**2 + d*x + 1

popt1, pcov1 = curve_fit(func_polinomial, x1, y1)
pred_x = list(range(min(x1),max(x1)+7))

pred_x = np.array(pred_x, dtype=float)
print(y1)
#grficamos los dos metodos para analizar


plt.plot(x, y,label='Predicion Probabilidad')
plt.plot(pred_x,func_polinomial(pred_x,*popt1),label='Predicion Matematica')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```

```
[1.0, 6.0, 7.0, 10.0, 13.0, 14.0, 15.0, 17.0, 23.0, 28.0, 37.0, 58.0, 11
1.0, 168.0, 199.0, 426.0, 532.0, 789.0, 981.0, 1082.0, 1211.0, 1403.0, 1
627.0, 1835.0, 1890.0, 1966.0, 2302.0, 2758.0, 3163.0, 3368.0, 3465.0, 3
646.0, 3747.0, 3995.0, 4450.0, 4965.0, 7161.0, 7257.0, 7466.0, 7529.0, 7
603.0, 7858.0, 8225.0, 8450.0, 9022.0, 9468.0, 10128.0, 10398.0, 10850.
0, 11183.0, 11183.0, 11183.0, 22719.0, 23240.0, 24258.0, 24675.0, 24934.
0, 26336.0, 27464.0, 29538.0, 31881.0, 31881.0, 29420.0, 30298.0, 28818.
0, 29071.0, 29559.0, 29509.0, 30419.0, 30486.0, 30502.0, 31467.0, 32763.
0, 33182.0, 33582.0, 34151.0, 34854.0, 35306.0, 35828.0, 36258.0, 36756.
0, 37355.0, 37355.0, 38103.0, 38471.0, 38571.0, 38571.0, 39098.0, 39994.
0, 40414.0, 40966.0, 40966.0, 41575.0, 42728.0, 43120.0, 43378.0, 43917.
0, 44440.0, 44440.0, 45778.0, 46356.0, 46751.0, 47322.0, 47943.0, 48490.
0, 49097.0, 49731.0, 50183.0, 50640.0, 50640.0, 51643.0, 52334.0, 53156.
0, 53856.0, 54574.0, 55255.0, 55665.0, 56432.0, 58257.0, 59468.0, 60657.
0, 61535.0, 61958.0, 62380.0, 63245.0, 64221.0, 65018.0, 65801.0, 67209.
0, 67870.0, 68459.0, 69570.0, 70329.0, 71365.0, 72444.0, 73382.0, 74013.
0, 74620.0, 76217.0, 77257.0, 78148.0, 79049.0, 80036.0, 80694.0, 81161.
0, 82279.0, 83193.0, 84370.0, 85355.0, 86232.0, 86524.0, 87041.0, 87963.
0, 88866.0, 90537.0, 91969.0, 93572.0, 94459.0, 94701.0, 95563.0, 97110.
0, 98343.0, 99409.0, 100688.0, 101542.0, 101751.0, 102941.0, 104475.0, 1
05508.0, 106481.0, 107089.0, 107769.0, 108289.0, 109030.0, 110549.0, 111
219.0, 112141.0, 112906.0, 113648.0, 113767.0, 114309.0, 115457.0, 11636
0.0, 117175.0, 118045.0, 109784.0, 110092.0, 110757.0, 112166.0, 113206.
0, 114732.0, 116451.0, 118594.0, 118911.0, 119553.0, 121525.0, 122257.0,
124129.0, 125620.0, 126419.0, 126711.0, 127643.0, 129892.0, 131146.0, 13
2475.0, 133981.0, 134747.0, 134965.0, 135749.0, 137047.0, 138584.0, 1395
34.0, 140351.0, 141034.0, 141339.0, 142056.0, 143531.0, 145045.0, 14584
8.0, 146828.0, 147033.0, 147315.0, 148171.0, 149083.0, 150360.0, 151659.
0, 152422.0, 153289.0, 153423.0, 154115.0, 155625.0, 156451.0, 158270.0,
159614.0, 161635.0, 162178.0, 163192.0, 164908.0, 166302.0, 167147.0, 16
8192.0, 169194.0, 169562.0, 170222, 170882, 171542, 172202, 172862, 1735
22, 174182, 174842]
[1.00000e+00 6.00000e+00 7.00000e+00 1.00000e+01 1.30000e+01 1.40000e+01
 1.50000e+01 1.70000e+01 2.30000e+01 2.80000e+01 3.70000e+01 5.80000e+01
 1.11000e+02 1.68000e+02 1.99000e+02 4.26000e+02 5.32000e+02 7.89000e+02
 9.81000e+02 1.08200e+03 1.21100e+03 1.40300e+03 1.62700e+03 1.83500e+03
 1.89000e+03 1.96600e+03 2.30200e+03 2.75800e+03 3.16300e+03 3.36800e+03
 3.46500e+03 3.64600e+03 3.74700e+03 3.99500e+03 4.45000e+03 4.96500e+03
 7.16100e+03 7.25700e+03 7.46600e+03 7.52900e+03 7.60300e+03 7.85800e+03
 8.22500e+03 8.45000e+03 9.02200e+03 9.46800e+03 1.01280e+04 1.03980e+04
 1.08500e+04 1.11830e+04 1.11830e+04 1.11830e+04 2.27190e+04 2.32400e+04
 2.42580e+04 2.46750e+04 2.49340e+04 2.63360e+04 2.74640e+04 2.95380e+04
 3.18810e+04 3.18810e+04 2.94200e+04 3.02980e+04 2.88180e+04 2.90710e+04
 2.95590e+04 2.95090e+04 3.04190e+04 3.04860e+04 3.05020e+04 3.14670e+04
 3.27630e+04 3.31820e+04 3.35820e+04 3.41510e+04 3.48540e+04 3.53060e+04
 3.58280e+04 3.62580e+04 3.67560e+04 3.73550e+04 3.73550e+04 3.81030e+04
 3.84710e+04 3.85710e+04 3.85710e+04 3.90980e+04 3.99940e+04 4.04140e+04
 4.09660e+04 4.09660e+04 4.15750e+04 4.27280e+04 4.31200e+04 4.33780e+04
 4.39170e+04 4.44400e+04 4.44400e+04 4.57780e+04 4.63560e+04 4.67510e+04
 4.73220e+04 4.79430e+04 4.84900e+04 4.90970e+04 4.97310e+04 5.01830e+04
 5.06400e+04 5.06400e+04 5.16430e+04 5.23340e+04 5.31560e+04 5.38560e+04
 5.45740e+04 5.52550e+04 5.56650e+04 5.64320e+04 5.82570e+04 5.94680e+04
 6.06570e+04 6.15350e+04 6.19580e+04 6.23800e+04 6.32450e+04 6.42210e+04
 6.50180e+04 6.58010e+04 6.72090e+04 6.78700e+04 6.84590e+04 6.95700e+04
 7.03290e+04 7.13650e+04 7.24440e+04 7.33820e+04 7.40130e+04 7.46200e+04
 7.62170e+04 7.72570e+04 7.81480e+04 7.90490e+04 8.00360e+04 8.06940e+04
 8.11610e+04 8.22790e+04 8.31930e+04 8.43700e+04 8.53550e+04 8.62320e+04
 8.65240e+04 8.70410e+04 8.79630e+04 8.88660e+04 9.05370e+04 9.19690e+04
 9.35720e+04 9.44590e+04 9.47010e+04 9.55630e+04 9.71100e+04 9.83430e+04
 9.94090e+04 1.00688e+05 1.01542e+05 1.01751e+05 1.02941e+05 1.04475e+05
 1.05508e+05 1.06481e+05 1.07089e+05 1.07769e+05 1.08289e+05 1.09030e+05
```

```
1.10549e+05 1.11219e+05 1.12141e+05 1.12906e+05 1.13648e+05 1.13767e+05
1.14309e+05 1.15457e+05 1.16360e+05 1.17175e+05 1.18045e+05 1.09784e+05
1.10092e+05 1.10757e+05 1.12166e+05 1.13206e+05 1.14732e+05 1.16451e+05
1.18594e+05 1.18911e+05 1.19553e+05 1.21525e+05 1.22257e+05 1.24129e+05
1.25620e+05 1.26419e+05 1.26711e+05 1.27643e+05 1.29892e+05 1.31146e+05
1.32475e+05 1.33981e+05 1.34747e+05 1.34965e+05 1.35749e+05 1.37047e+05
1.38584e+05 1.39534e+05 1.40351e+05 1.41034e+05 1.41339e+05 1.42056e+05
1.43531e+05 1.45045e+05 1.45848e+05 1.46828e+05 1.47033e+05 1.47315e+05
1.48171e+05 1.49083e+05 1.50360e+05 1.51659e+05 1.52422e+05 1.53289e+05
1.53423e+05 1.54115e+05 1.55625e+05 1.56451e+05 1.58270e+05 1.59614e+05
1.61635e+05 1.62178e+05 1.63192e+05 1.64908e+05 1.66302e+05 1.67147e+05
1.68192e+05 1.69194e+05 1.69562e+05 1.70222e+05 1.70882e+05 1.71542e+05
1.72202e+05 1.72862e+05 1.73522e+05 1.74182e+05 1.74842e+05]
```

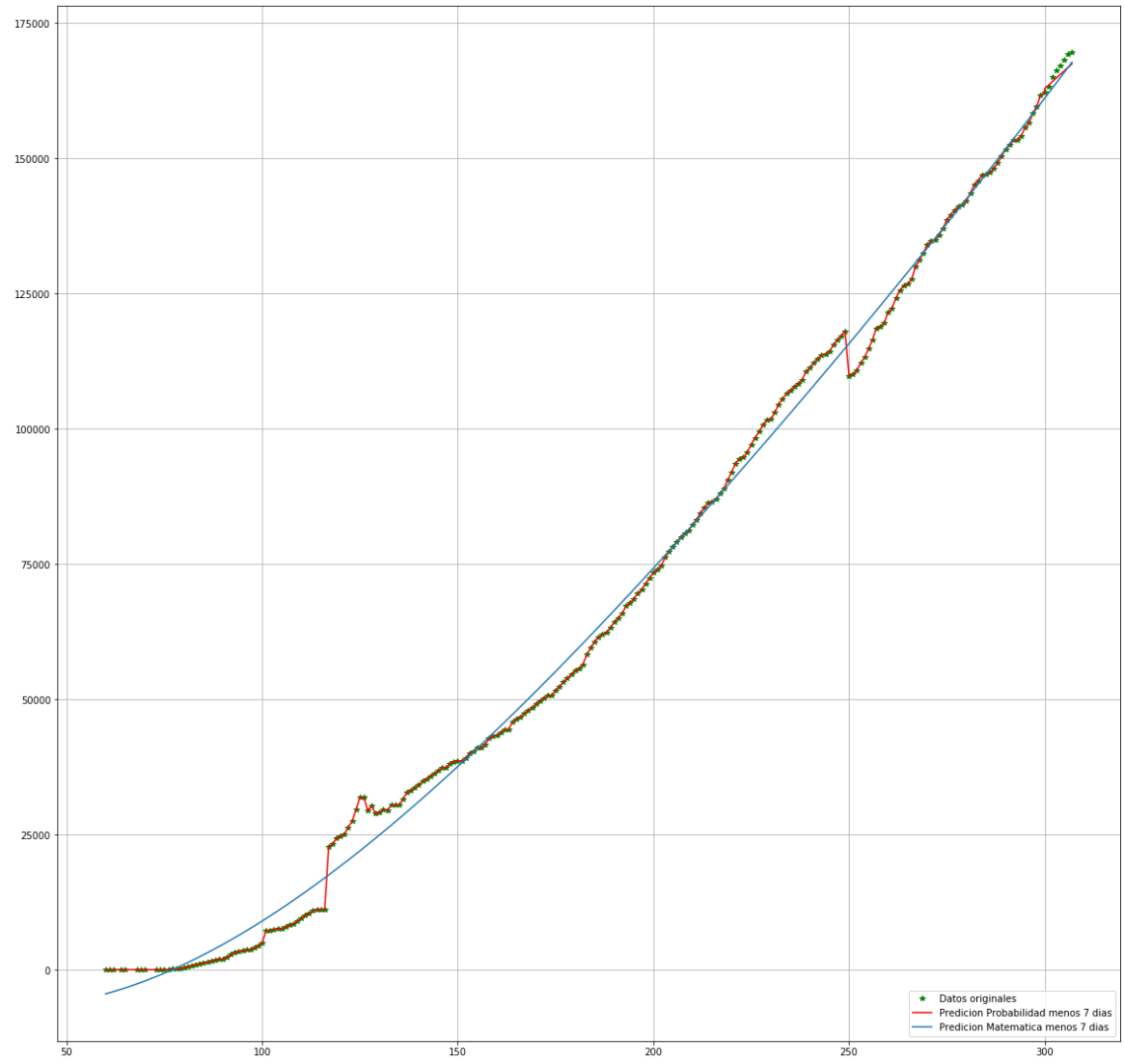


2. Retroceder un semana y comparar el modelo matematico vs probabilidad vs reales. Solo cargan los datos para generar los modelos menos 7 dias.

In [83]:

```
df2 = pd.read_csv('datacovid.csv')

df2 = df2[df2['location'].isin(['Ecuador'])] #Filtro la Informacion solo para Ecuado
r
df2 = df2[(df2.total_cases >= 1)]
df2 = df2.loc[:,['date', 'new_cases', 'total_cases']] #Selecciono las columnas de an
alasis
FMT = '%Y-%m-%d'
date = df2['date']

df2['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("20
20-01-01", FMT)).days)

data = df2[:-7]

#Modelo probablistico
filtro1 = data["new_cases"] # Filtro los datos que se empezo a tener casos
#Obtenemos la mediana
media1 = filtro1.mean()
mediana1 = filtro1.median()

y1 = list(data.iloc [:, 2]) # Total casos
x1 = list(data.iloc [:, 0]) # Dias

#Realizamos un ejemplo de prediccion
prediccion_siguiente1 = int(y1[-1] + mediana1)
print(prediccion_siguiente1)

for i in range(x1[-1], x1[-1]+8):
    x1.append(i)
    y1.append(int(y1[-1] + mediana))


#modelo matematico
x1 = np.array(x1, dtype=float)
y1 = np.array(y1, dtype=float)
def func_polinomial(x, a, b, c, d):
    return a*x**4 + b*x**3 + c*x**2 + d*x + 1

popt1, pcov1 = curve_fit(func_polinomial, x1, y1)

x2 = df2.date
y2 = df2.total_cases

plt.figure(figsize=(20,20))
plt.plot(x2, y2,'g*',label='Datos originales')
plt.plot(x1, y1,color='r',label='Predicion Probabilidad menos 7 dias')
plt.plot(x1,func_polinomial(x1,*popt1),label='Predicion Matematica menos 7 dias')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```

162816


175000
150000
125000
100000
75000
50000
25000
0
50
100
150
200
250
300
Datos originales
Predicion Probabilidad menos 7 dias
Predicion Matematica menos 7 dias

Analisis

Los dos modelos tanto el matemático como el de probabilidad devuelven una predicción casi idéntica, pero el modelo de probabilidad es mucho más fácil de implementar lo que nos permite ahorrar tiempo en el desarrollo de la simulación.

En el segundo punto, al cargar la información con menos 7 días, de los datos originales y luego hacer uso de los dos métodos tanto matemático como el de probabilidad y prediciendo 7 días más podemos observar que ambos nos dan una aproximación similar en datos y al comparar y graficar los datos originales se observa que con la proyección se obtiene un número más bajo de infectados que son los que se tiene en la actualidad, el número de infectados es de 162816 mil casos.

Observando esto podríamos decir que los datos proporcionados para la simulación no son del todo confiables y al parecer no están apegados a la realidad por lo que nuestras soluciones de simulación al realizar una tarea correcta la solución no se apega a los datos reales.

Conclusiones

Estos modelos deben ser usados como referencia y con mas datos se puede lograr una predicción mucho mayor, se recomienda el investigar sobre nuevas técnicas o combinaciones que mejoren los resultados del modelo.

Referencias

- https://www.researchgate.net/publication/340092755_Infeccion_del_Covid-19_en_Colombia_Una_comparacion_de_modelos_logisticos_y_exponenciales_aplicados_a_la_infeccior (https://www.researchgate.net/publication/340092755_Infeccion_del_Covid-19_en_Colombia_Una_comparacion_de_modelos_logisticos_y_exponenciales_aplicados_a_la_infeccior
- https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/ (https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/)