



Simulación

Tema: Predicción basados en modelos matemáticos y probabilísticos.

Prueba Practica 1

Nombre: Xavier Jarro

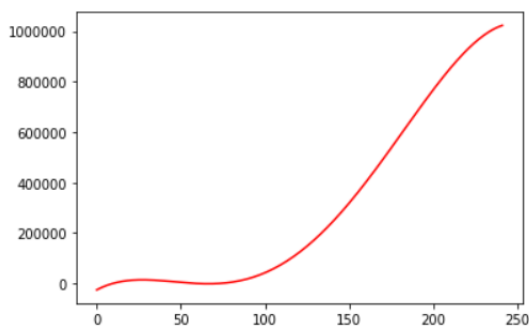
Materia: Simulación

- Diseñe y desarrolle un modelo y/o *script* que permita simular el siguiente caso real:
 - Investigar los datos de los países contagiados por COVID-19, especialmente de Latinoamérica (menos Ecuador), deberán escoger uno y que no se repita, para ello se va a seleccionar el orden en el que publique dentro del foro “Tema prueba 1”, con estos datos obtener los siguientes modelos:

Carga de datos y representacion.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.linear_model import LinearRegression
4 import pandas as pd
5 from datetime import datetime, timedelta
6 import matplotlib.pyplot as plt
7 from pylab import *
8 import sympy as sp
9 import matplotlib.pyplot as plt
10 from scipy.optimize import curve_fit
11 from sklearn.linear_model import LogisticRegression
12 from sklearn.metrics import mean_squared_error
13 from scipy.optimize import fsolve
14 from sklearn import linear_model
15 %matplotlib inline
16
17
18
19 df = pd.read_csv('covid.csv').fillna(0)
20 ndf= df.loc[(df['Country'] == 'Colombia') & (df['Cumulative_cases'] != 0)]
21 ndf1=ndf[['Date_reported', 'Cumulative_cases']]
22 x=np.arange(1,len(ndf1)+1,1, dtype='float')
23 y=np.array(ndf1.values[:,1], dtype='float')
24 x2=np.arange(1,len(ndf1)+1,1)
25 y2=np.array(ndf1.values[:,1])
```

```
1 fun1 = np.poly1d(np.polyfit(x, y, 4))
2 plt.plot(fun1(x), c='red')
3 plt.show()
```



- Generar un modelo matematico de prediccion para regresion lineal, exponencial, polinomico y logaritmico, de los nuevos contactos en la proxima semana (7 dias despues).



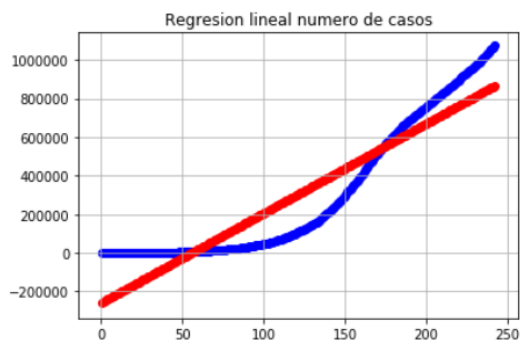
Simulación

Tema: Predicción basados en modelos matemáticos y probabilísticos.

Prueba Practica 1

Regresion lineal.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn import linear_model
4
5 modelo = linear_model.LinearRegression()
6 modelo.fit(x.reshape((-1,1)), y)
7 y_pred = modelo.predict(x.reshape((-1,1)))
8
9
10 plt.scatter(x,y,color='blue')
11 plt.title("Regresion lineal numero de casos")
12 plt.scatter(x,y_pred,color='red')
13 plt.grid(True)
14 plt.show()
15
16 if (modelo.intercept_ < 0):
17     ecua='y = {}x {}'
18 else:
19     ecua='y = {}x + {}'
20 print("La ecuacion es la siguiente: ", ecua.format(modelo.coef_[0],modelo.intercept_))
```

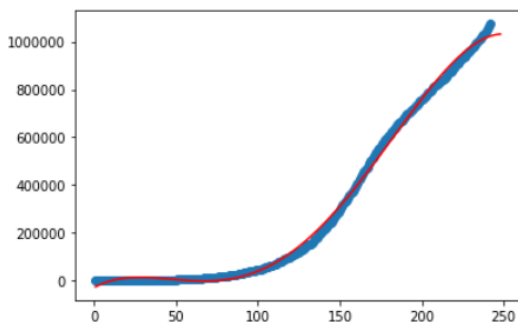


La ecuacion es la siguiente: $y = 4645.943122071123x - 259643.52734817035$

Regresion Polinomial

```
1 fun1 = np.poly1d(np.polyfit(x, y, 4))
2 print(fun1)
3 plt.scatter(x, y)
4 x1=np.arange(1,len(ndf1)+7,1, dtype='float')
5 plt.plot(x1, fun1(x1), c='r')
6 plt.show()
```

$-0.001964 x^4 + 0.9034 x^3 - 100.9 x^2 + 3712 x - 2.922e+04$





Simulación

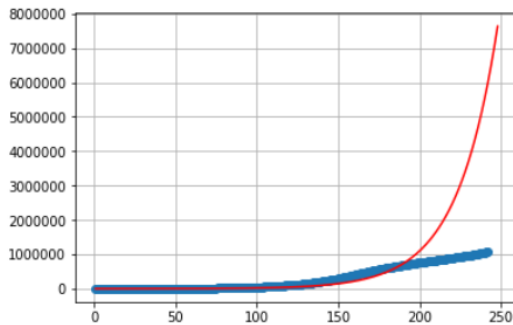
Tema: Predicción basados en modelos matemáticos y probabilísticos.

Prueba Practica 1

Regresion Exponencial. ¶

```
1 curve_fit=np.polyfit(x2, np.log(y), deg=1)
2 print(curve_fit)
3 pred_x=np.array(list(range(min(x2), max(x2)+7)))
4 yx=np.exp(curve_fit[1])*np.exp(curve_fit[0]*pred_x)
5 plt.plot(x2,y2,'o')
6 plt.plot(pred_x,yx, color="red")
7 plt.grid(True)
```

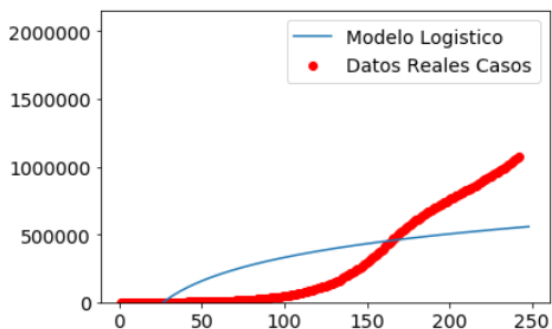
[0.040454 5.81575841]



Regresion Logaritmica.

```
1 def modelo_logistico(x2,a,b):
2     return a+b*np.log(x2)
3
4 exp_fit = curve_fit(modelo_logistico,x2,y2)
```

```
1 pred_x = list(range(min(x2),max(x2)+7))
2 plt.rc('font', size=14)
3 plt.scatter(x2,y2,label="Datos Reales Casos",color="red")
4 plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], label="Modelo Logistico" )
5 plt.legend()
6 plt.ylim(0,max(y)*2)
7 plt.show()
```



- Generar un modelo probabilístico con los datos.



Simulación

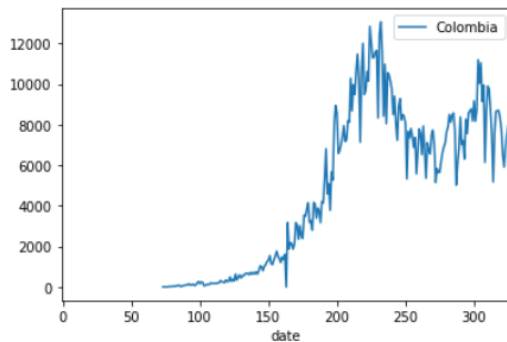
Tema: Predicción basados en modelos matemáticos y probabilísticos.

Prueba Practica 1

Regresion probabilistica.

```
1 import pandas as pd
2 import numpy as np
3 from datetime import datetime, timedelta
4 from sklearn.metrics import mean_squared_error
5 from scipy.optimize import curve_fit
6 from scipy.optimize import fsolve
7 from sklearn import linear_model
8 import matplotlib.pyplot as plt
9 %matplotlib inline
10
11 url = 'https://covid.ourworldindata.org/data/ecdc/new_cases.csv'
12 df = pd.read_csv(url)
13
14 df = df.loc[:, ['date', 'Colombia']] #Selecciono las columnas de analisis
15 # Expresar las fechas en numero de dias desde el 01 Enero
16 FMT = '%Y-%m-%d'
17 date = df['date']
18 df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01-01", FMT)).days)
```

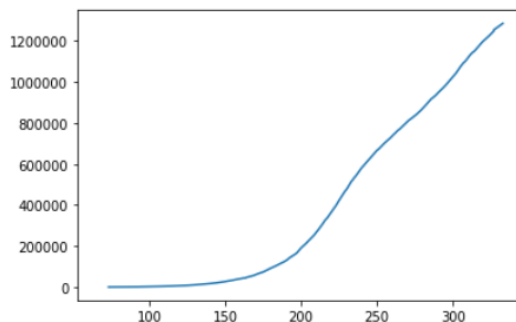
```
1 df.plot(x='date', y='Colombia')
```



```
1 filtro = df["Colombia"][61:]
2 media = filtro.mean()
3 mediana = filtro.median()
```

```
1 url = 'https://covid.ourworldindata.org/data/ecdc/total_cases.csv'
2 df_t = pd.read_csv(url)
3 FMT = '%Y-%m-%d'
4 date = df_t['date']
5 df_t['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01-01", FMT)).days)
6 df_t = df_t.loc[:, ['date', 'Colombia']]
7 y = list(df_t.iloc[:, 1])
8 x = list(df_t.iloc[:, 0])
9 prediccion_siguiete = int(y[-1] + mediana)
```

```
1 for i in range(x[-1], x[-1]+7):
2     x.append(i)
3     y.append(int(y[-1] + mediana))
4 plt.plot(x[61:], y[61:])
5 plt.show()
```





Simulación

Tema: Predicción basados en modelos matemáticos y probabilísticos.

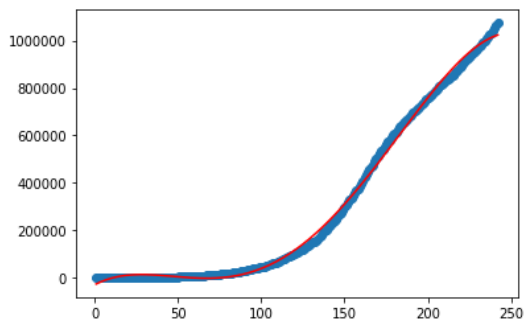
Prueba Practica 1

- El proceso de simulación desarrollado deberá considerar los siguientes aspectos:
 - Se debe establecer un modelo basado en modelos matemáticos y probabilísticos.

Modelo matematico. ¶

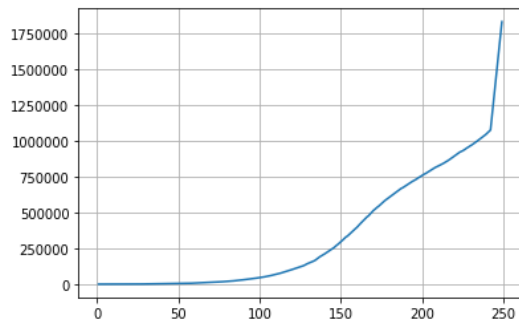
```
1  %matplotlib inline
2  from pylab import *
3  import numpy as np
4  import pandas as pd
5  import sympy as sp
6
7
8  df = pd.read_csv('covid.csv').fillna(0)
9  ndf= df.loc[(df[' Country'] == 'Colombia') & (df[' Cumulative_cases'] != 0)]
10 ndf1=ndf[['Date_reported',' Cumulative_cases']]
11 x=np.arange(1,len(ndf1)+1,1, dtype='float')
12 y=np.array(ndf1.values[:,1], dtype='float')
13
14 crx=x
15 cry=y
16
17 fun1 = np.poly1d(np.polyfit(x, y, 4))
18 print(fun1)
19 y_pred=fun1(x)
20 plt.scatter(x, y)
21 plt.plot(x, y_pred, c='r')
22 plt.show()
23
24
```

$$-0.001964 x^4 + 0.9034 x^3 - 100.9 x^2 + 3712 x - 2.922e+04$$



Modelo probabilistico.

```
1  media = ndf1.values[:,1].mean()
2  mediana = np.median(ndf1.values[:,1])
3
4  for it in range(int(x[-1]), int(x[-1]+7)):
5      x=np.append(x, (it+1))
6      y=np.append(y, y[-1] + mediana)
7  plt.plot(x,y)
8  plt.grid(True)
9  plt.show()
```





Simulación

Tema: Predicción basados en modelos matemáticos y probabilísticos.

Prueba Practica 1

Comparacion de modelos.

```
1 y_pred=np.array([])
2 y_pred=fun1(x)
3
4 from tabulate import tabulate
5 table=[]
6 print(tabulate(table))
7 for i,j,k in zip(x,y,y_pred):
8     table.append([i,j,round(k,2)])
9 print(tabulate(table, headers=["Dia", "M.Probabilistico", "M.Matematico"]))
```

Dia	M.Probabilistico	M.Matematico
1	5	-25607.2
2	5	-22191.3
3	5	-18966.6
4	5	-15927.8
5	7	-13069.5
6	7	-10386.7
7	13	-7874.03
8	20	-5526.51
9	28	-3339.04
10	28	-1306.6
11	38	575.77
12	49	2313.02
13	97	3910.01
14	108	5371.58
15	145	6702.51
16	196	7907.53
17	196	8991.34
18	196	9958.58
19	306	10813.9
20	306	11561.7
21	470	12206.6
22	470	12753
23	491	13205.4
24	539	13568
25	608	13845.3
26	702	14041.5
27	798	14160.7
220	894300	919185
221	902747	925910
222	911316	932480
223	919083	938890
224	924098	945134
225	930159	951208
226	936982	957106
227	945354	962823
228	952371	968355
229	959572	973695
230	965883	978838
231	974139	983778
232	981700	988512
233	990270	993032
234	998942	997333
235	1.00771e+06	1.00141e+06
236	1.01588e+06	1.00526e+06
237	1.02505e+06	1.00887e+06
238	1.03322e+06	1.01224e+06
239	1.04194e+06	1.01536e+06
240	1.05312e+06	1.01823e+06
241	1.06315e+06	1.02085e+06
242	1.07418e+06	1.02319e+06
243	1.18199e+06	1.02527e+06
244	1.2898e+06	1.02707e+06
245	1.39761e+06	1.02859e+06
246	1.50541e+06	1.02981e+06
247	1.61322e+06	1.03075e+06
248	1.72103e+06	1.03138e+06
249	1.82884e+06	1.03171e+06



Simulación

Tema: Predicción basados en modelos matemáticos y probabilísticos.

Prueba Practica 1

Mejor predicción.

El modelo matemático tuvo mejor resultado de predicción ya que se da un dato más real de lo que podría suceder a futuro.

Ventajas y desventajas.

Ventajas M. probabilístico:

- * Cálculo rápido de medidas y varianzas.
- * El muestreo probabilístico es sencillo y de fácil comprensión.

Desventajas M. probabilístico.

- * Existe un margen de error.
- * Si trabajamos con muestras pequeñas es posible que no se represente correctamente a toda la población adecuadamente, es por esto que se debe manejar un muestreo completo.

Ventajas M. matemático:

- * Se adecua a una predicción de una situación real.
- * Permiten una identificación rápida de las expectativas esperadas

Desventajas M. matemático.

- * Se pierde información del fenómeno que se está estudiando.
- * La recolección de datos puede ser muy costosa y complicada.