

## ▼ Universidad Politecnica Salesiana

### Examen Interciclo de Simulación

Realizado por: Damian Wilfrido Sumba

#### **Introducción:**

El golpe económico de la crisis sanitaria del corona virus no va a ser cosa de semanas, sino de meses. Dentro de una de las etapas importantes que están a la vuelta de la esquina son las elecciones presidenciales y asambleístas del Ecuador. Para ello se plantea realizar un sistema de regresión que permita identificar cual es la tendencia de los votos en base al manejo de las redes sociales (Twitter y/o Facebook) [1].

Las regresiones lineales pueden aprender por sí mismos y en este caso obtener automáticamente esa “recta” que buscamos con la tendencia de predicción. Para hacerlo se mide el error con respecto a los puntos de entrada y el valor “Y” de salida real [3].

#### **Enunciado:**

Diseñe y desarrolle un modelo y/o script que permita simular el siguiente caso real: ◦ Obtener datos de tendencia de twitter o facebook, para ello se puede obtener a través del API [4]. \* Title: Título del Post/Twitter

```
Word count: la cantidad de palabras del artículo,  
"#" of Links: los enlaces externos que contiene,  
"#" of comments: cantidad de comentarios,  
"#" Shares: compartidos.  
HashTag  
Etc.
```

En base a ello, se pretende proponer y generar una predicción de cuántas veces será compartido un post/twitter utilizando regresión [2].

Posteriormente se debe seguir un procesos de votación de eventos discretos que se describe a continuación: ◦ Tomar los resultados de la regresión para la selección del candidato. ◦ Se tiene una tendencia del 90% de personas que realizan el proceso de elección dentro del Ecuador. ◦ Dentro del procesos se tiene que alrededor del 5% - 10% votan nulo. ◦ Solo se va a tener en cuenta las elecciones de los asambleístas por el Azuay. ◦ Las personas solo tiene un recinto electoral para realizar el proceso. ◦ Las personas solo pueden realizar un proceso de elección por asambleísta del Azuay. ◦ La persona se acerca a la mesa electoral y hacen fila en caso de ser necesario. ◦ Realiza el voto en un tiempo aleatorio de un partido específico. ◦ La persona recibe su certificado votación. ◦ La persona sale del recinto electoral. ◦ Finalmente generar una grafica de las personas que votaron y los asambleístas electos. El proceso de simulación desarrollado deberá considerar los siguientes aspectos: ◦ Se debe establecer un modelo basado en modelos matemáticos y probabilísticos para la predicción del numero de veces que se compartirá o la tendencia electoral basada en redes sociales. ◦ El programa deberá generar gráficas que indiquen la ecuación matemática y probabilística de tendencias de votaciones. ◦ Deben calcularse las siguientes métricas del sistema de simulación de eventos discretos :

- Total de de personas que realizaron el proceso de votación.
- Asambleístas ganadores.
- El tiempo promedio de espera.

El informe en pdf (Archivo Notebook) se deberá subir al Git personal, todo el sistema debe ser basado en Python - Open Source. Cualquier adición o librerías que no se revisaron en clases serán tomada como puntos adicionales al interciclo. Referencias: [1] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/>

[2] <https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>

[3] <https://eprints.ucm.es/48804/1/TFM%20Manuel%20Alejandro%20Rodriguez%20Santana.pdf>

[4] [http://opac.pucv.cl/pucv\\_txt/txt-8000/UCC8094\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-8000/UCC8094_01.pdf)

## Desarrollo:

A conti niacion ontendremos infirmacion de las Red social de **Twitter** esta informacion la obtendremos mediante la conexion a la API de esta Red social.

```
#Importamos Librerias
from time import sleep
from datetime import datetime
from textblob import TextBlob
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import numpy as np
import tweepy as tw
from IPython.display import display
import seaborn as sns
from tweepy import OAuthHandler
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn import linear_model
```

```
# credenciales para el uso y conexion con la API de twitter
consumer_key = '10L9N58KFd4giyFKUTbrsMHPJ'
consumer_secret = 'BWvfxkGw9zGkgHbw06e06hwT7zPicZN6Mdn9MMGTqS6JDBP8AB'
access_token = '1308637223793942528-XkR0mf8FbEUEKqqnvBWL04ACck1CgC'
access_token_secret = 'RWj0vB7bfcKohQ4jxs0Ck22hIUTdFd4kcjb7HRs5gpVpB'
```

```
#Se autentica en twitter
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

#se verifica que el usuario conectado en twitter es de uno
api = tw.API(auth)
print(api.me().name)
```

Damian Sumba

```
#en esta parte pondremos la informacion que estamos buscando, en este caso buscare CNE(consejo nacional electoral)
buscarTweet = "cne"
ts = api.search(buscarTweet, count=50, result_type='popular')
```

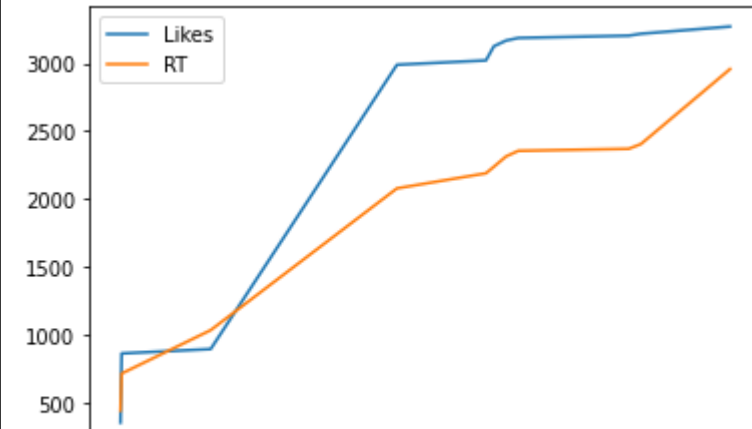
```
df = pd.DataFrame({'ID': list(map(lambda x: x.id , ts)),
                  'Usuario': list(map(lambda x: x.author.name , ts)),
                  'Fecha': list(map(lambda x: x.created_at, ts)),
                  'Texto': list(map(lambda x: x.text, ts)),
                  'RT': list(map(lambda x: x.retweet_count, ts)),
                  'Likes': list(map(lambda x: x.favorite_count, ts)),
                  'Followers': list(map(lambda x: x.author.followers_count, ts)),
                  '#Palabras': list(map(lambda x: len(x.text), ts)),
                  'Idioma': list(map(lambda x: x.lang, ts)),
                  'HashTag': list(map(lambda x: x[0]['text'] if x != [] else '', list(map(lambda x: x.entities['hashtags'],
                                                                                          ts)))),
                  })
df['Fecha'] =pd.to_datetime(df.Fecha)
df.sort_values(by='Followers')
df
```



	ID	Usuario	Fecha	Texto	RT	Likes	Followers	#Palabras	Idioma	
0	1344299880743899136	Gustavo Rugeles	2020-12-30 15:10:42	Los magistrados del CNE Hernán Penagos y Jorge...	446	357	80333	140	es	
1	1344293431728857097	José Miguel Santamaría Uribe	2020-12-30 14:45:04	independientemente de las nuevas trabas para l...	272	509	23182	133	es	RevocatoriaCl
2	1344412951231385602	VTV CANAL 8	2020-12-30 22:40:00	Presidente del TSJ recibe informe sobre celebr...	319	32	1788945	134	es	GMVVCumpli
3	1344072968788439040	Rafael Correa	2020-12-30 00:09:02	●●● La campaña "empieza" el 31, pero ya el pue...	1043	2091	3742857	140	es	
4	1344239307091898372	VTV CANAL 8	2020-12-30 11:10:00	Presidente del TSJ recibe informe sobre celebr...	111	32	1788945	120	es	
5	1344529280483667971	Manuel Zelaya R.	2020-12-31 06:22:15	El CNE confirma mis denuncias y por unanimidad...	48	104	158411	139	es	

```
# en la siguiente grafica podremos ver cuantas veces q se retwitteado y dado un me gusta a las twits con la palabra CNE
df1 = df.iloc[:, 2:7].cumsum()
df1.plot(x='Followers', y=['Likes','RT'])
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f11b21a0198>



```
df1["sum"] = df1[['Likes','Followers']].sum(axis=1)
df2 = df1.loc[:, ['RT','Likes','sum']]
df2
```

	RT	Likes	sum
0	446	357	80690
1	718	866	104381
2	1037	898	1893358
3	2080	2989	5638306
4	2191	3021	7427283
5	2239	3125	7585798
6	2315	3167	7831110
7	2356	3186	8076399
8	2371	3203	10295932
9	2406	3217	10541216
10	2957	3270	12330214

```
# Con los datos obtenidos ahora aplicaremos regresion lineal para sacar un pronostico de cuantas veces mas se va a compartit
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = list(df2.iloc[:, 0]) # likes
y = list(df2.iloc[:, 1]) # Total
```

```
def promedio(x,y):
    return sum(x) / len(y)
```

```
def operacion1(x,y):
    #obtiene x menos el promedio de x
    a = x-np.average(x)
    b = y-np.average(y)
    promxy = sum(a*b)
    promxx = sum(a*a)
    result =promxy/promxx
    return result
```

```
def operacion0(x,y):
    res = np.average(y)-operacion1(x,y)*np.average(x)
    return res
```

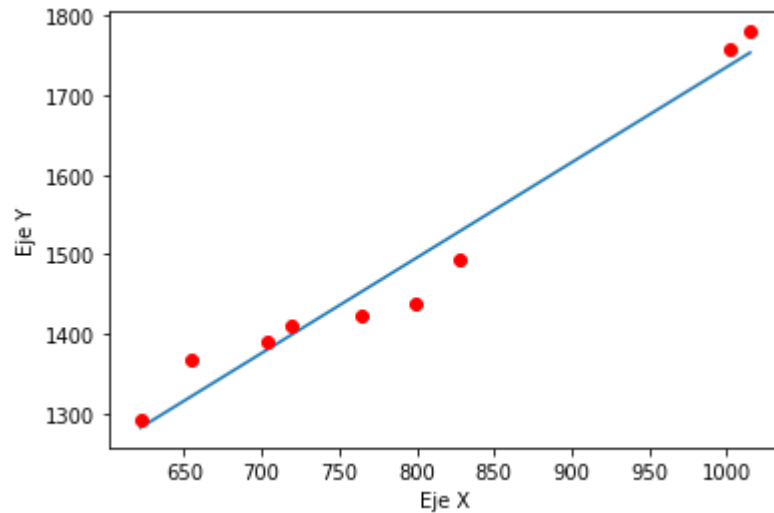
```
z = api.get_user('yakuperezg').followers_count
```

```
def graficar(x,y,z):
    b1 =operacion1(x,y)
    b0 =operacion0(x,y)
    predecir= b1*z+b0
    puntos_x = np.linspace(x[0],x[-1],6)
    puntos_y = b0+b1*puntos_x
    print("Ecuación de la recta:",puntos_x, 'Z=',b1,'*',z,'+',b0)
    print('Prediccion: ',predecir)
    plt.plot(puntos_x,puntos_y,)
    plt.plot(x,y,"o", color='red')
    plt.xlabel('Eje X')
    plt.ylabel('Eje Y')
```

```
plt.show()
```

```
if __name__=="__main__":  
    graficar(x,y,z)
```

Ecuación de la recta: [ 622. 700.6 779.2 857.8 936.4 1015. ] Z= 1.2002588836087664 \* 57207 + 535.1733172565431  
Predicción: 69198.38327186325

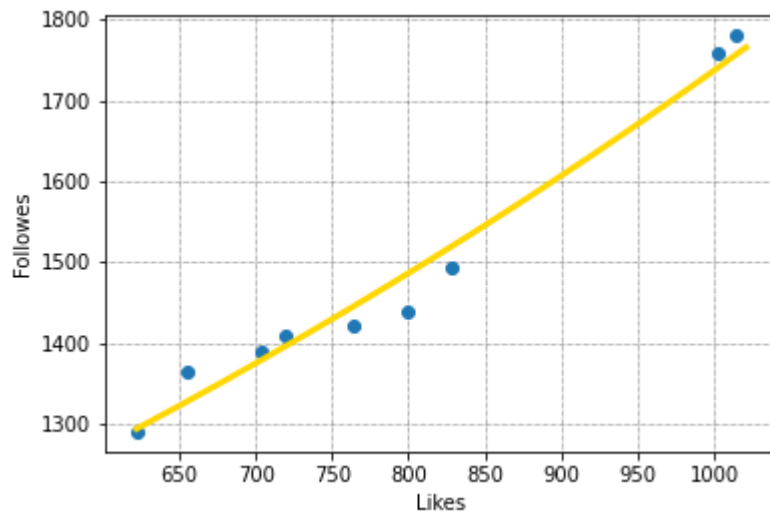


```
from scipy.optimize import curve_fit  
  
x = list(df2.iloc[:, 0]) # likes  
y = list(df2.iloc[:, 1]) # Total  
z = api.get_user('yakuperezg').followers_count  
  
curve_fit=np.polyfit(x,np.log(y),deg=1)  
# print(curve_fit)  
  
pred_x=np.array(list(range(min(x),max(x)+7)))  
yx=np.exp(curve_fit[1])*np.exp(curve_fit[0]*pred_x)  
plt.plot(x,y,"o")  
plt.plot(pred_x,yx,color='gold',linewidth=3.0)  
  
print('Predicción: ', str(z))
```



```
#Propiedades
plt.xlabel('Likes')
plt.ylabel('Followes')
plt.grid(color='black', linestyle='dotted', linewidth=0.5)
```

Predicción: 57208



```
# Creacion del modelo discreto para la simulacion de elecciones presidenciales.
```

```
import random
import pylab as py
import simpy
import numpy as np
import pandas as pd
```

```
def votacion():
```

```
    i = 0
```

```
    persona = []
```

```
    llegada = []
```

```
    espera = []
```

```
    votando = []
```

```
    papeletas = []
```

```
    salida = []
```

```
    mesa = 350
```

```
    lista_total = [['Persona:', 'Llego a la mesa:', 'Tiempo de espera:', 'Proceso de votación:', 'Voto finalizado:']]
```

```

lista_total=[[ Persona:  , Llega a las:  , Tiempo de espera , Proceso de votacion , voto finalizado ]]
while(i<mesa):
    if(i==0):
        persona.append(i)
        llegada.append(i)
        espera.append(random.random())
        votando.append(random.random())
        papeletas.append(i)
        #print(type(espera[i]))
        tiempo_salida =llegada[i] + espera[i] + votando[i]
        salida.append(tiempo_salida)
    if(i>=1):
        persona.append(i)
        llegada.append(llegada[i-1]+random.random())
        espera.append(max(salida[i-1],llegada[i])-llegada[i])
        votando.append(random.random())
        papeletas.append(max(salida[i-1],llegada[i])-salida[i-1])
        tiempo_salida =llegada[i] + espera[i] + votando[i]
        salida.append(tiempo_salida)
    i=i+1
i=0

while(i<mesa):
    lista=[str(persona[i]).format(2/3),str(llegada[i]).format(10/3).format(10/3),str(espera[i]).format(10/3),str(votando
    lista_total.append(lista)
    i=i+1
print(type(lista_total))
data=pd.DataFrame(lista_total)
print(data)
votacion()

```

```
<class 'list'>
```

	0	1	...	3	4
0	Persona:	Llega a las:	...	Proceso de votación	Voto finalizado
1	0	0	...	0.7394904694886657	0.9506160813017275
2	1	0.409659049444853	...	0.2726510620616295	1.223267143363357
3	2	1.1585737497232436	...	0.2188201697227219	1.442087313086079
4	3	2.028152857123353	...	0.9793260955843247	3.007478952707678
..	...	...	...	...	...
346	345	174.51935256862092	...	0.30606614789907505	180.48275271593832

347	346	175.47164962281553	...	0.7354050314282257	181.21815774736655
348	347	175.52751063470697	...	0.05128391808492827	181.26944166545147
349	348	176.3147517167828	...	0.4264106242352055	181.69585228968668
350	349	176.8077492728125	...	0.5659963205699913	182.26184861025666

[351 rows x 5 columns]

```
import pandas as pd
url = "https://raw.githubusercontent.com/DamianSumba/Simulacion_ups/master/shares.txt"
df = pd.read_table(url, header = None)
df.head()

Arauz_digits = [0, 0, 0, 0, 0, 0, 0, 0, 0]
yaku_win_pct = 73.13
Arauz_win_pct = 25.28
number_of_sims = 600
total_wards = 0
total_votos_Arauz = 0
total_votos_yaku = 0
total_votes = 0
for i in range(number_of_sims):
    my_input = open(shares)
    for line in my_input:
        total_wards += 1
        fields = line.strip().split()
        num_voters = int(fields[0])
        Arauz_votes_in_ward = 0
        yaku_votes_in_ward = 0
        for j in range(num_voters):
            random_num = random.random() * 100
            if random_num <= yaku_win_pct:
                yaku_votes_in_ward += 1
                total_votos_yaku += 1
            elif random_num <= (yaku_win_pct + Arauz_win_pct) :
                Arauz_votes_in_ward += 1
                total_votos_Arauz += 1
        total_votes += 1
    Arauz_digit = int(str(Arauz_votes_in_ward)[0])
    yaku_digit = int(str(yaku_votes_in_ward)[0])
```

```
Arauz_digits[Arauz_digit] += 1
yaku_digits[yaku_digit] += 1
if i % 100 == 0:
    print('Simulacion con ', i)
Arauz_win_pct = 100.0 * total_votos_Arauz / total_votes
yaku_win_pct = 100.0 * total_votos_yaku / total_votes
print('Yaku Perez gana con:', yaku_win_pct , '%')
print('Arauz pierde con:', Arauz_win_pct, '%')
```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.