

Nombre: Jorge Arévalo

Docente: Ing. Diego Quisi

Materia: Simulación

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 assembleístas 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.

Solución

In [97]:

```
1 import pandas as pd
2 import os
3 import matplotlib.pyplot as plt
4 from facebook_scraper import get_posts
5 import numpy as np
6 import seaborn as sb
7 %matplotlib inline
8 from mpl_toolkits.mplot3d import Axes3D
9 from matplotlib import cm
10 plt.rcParams['figure.figsize'] = (16, 9)
11 plt.style.use('ggplot')
12 from sklearn import linear_model
13 from sklearn.metrics import mean_squared_error, r2_score
```

In [98]:

```
1 def fun(texto):
2     cuenta=0
3     for carac in texto:
4         cuenta = cuenta + 1
5     return cuenta
```

In [99]:

```
1 posts = []
2
3 for post in get_posts('estebanalbornozv ', pages=10):
4     post['numWord']=fun(post['text'])
5     posts.append(post)
6
7 for post in get_posts('GloriaAstudilloLoor', pages=10):
8     post['numWord']=fun(post['text'])
9     posts.append(post)
10
11 fb_posts = pd.DataFrame(posts)
```

In [101]:

```

1 file = open("dato.txt","w")
2 for like in fb_posts['shares']:
3     file.write(str(like)+'\n')
4 file.close()
5 fb_posts.describe()

```

Out[101]:

	likes	comments	shares	numWord
count	76.000000	76.000000	76.000000	76.000000
mean	7.039474	0.276316	0.105263	290.868421
std	5.084462	0.960537	0.449561	246.061312
min	1.000000	0.000000	0.000000	0.000000
25%	4.000000	0.000000	0.000000	195.500000
50%	6.000000	0.000000	0.000000	250.500000
75%	10.000000	0.000000	0.000000	283.250000
max	28.000000	6.000000	2.000000	1932.000000

In [102]:

```

1 fb_posts.head()

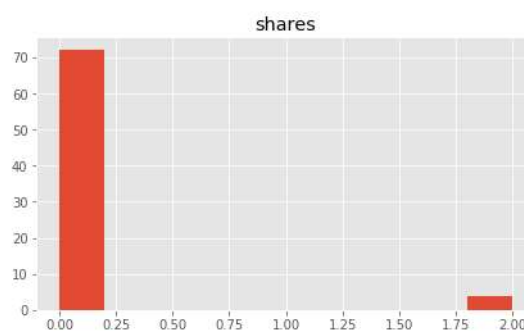
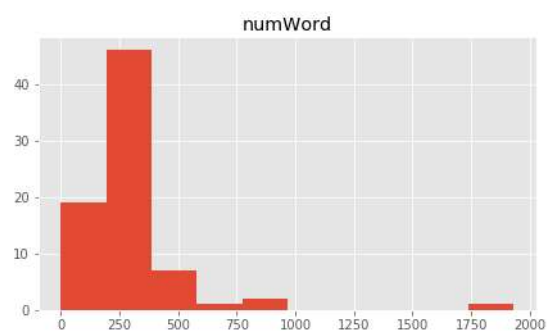
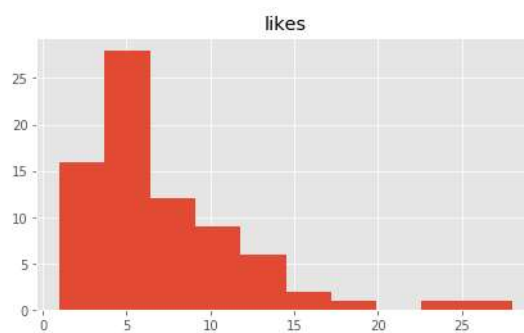
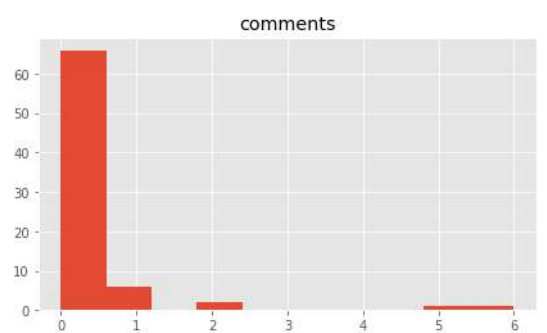
```

Out[102]:

	post_id	text	post_text	shared_text	time
0	1715171395311281	Culmina una exitosa semana de actividades en l...	Culmina una exitosa semana de actividades en l...		2020-12-20 16:38:20
1	1713716165456804	Fructífero diálogo con el medio de comunicació...	Fructífero diálogo con el medio de comunicació...		2020-12-18 22:51:40
2	1713236138838140	Desde la @DesarrolloEcAN, hemos entregado al p...	Desde la @DesarrolloEcAN, hemos entregado al p...		2020-12-18 09:45:57
3	1712738782221209	La #LeyPlásticosUnSoloUso nace de la necesidad...	La #LeyPlásticosUnSoloUso nace de la necesidad...		2020-12-17 18:31:15
4	1711696088992145	La #LeyPlásticosUnSoloUso establece obligacion...	La #LeyPlásticosUnSoloUso establece obligacion...		2020-12-16 10:58:00

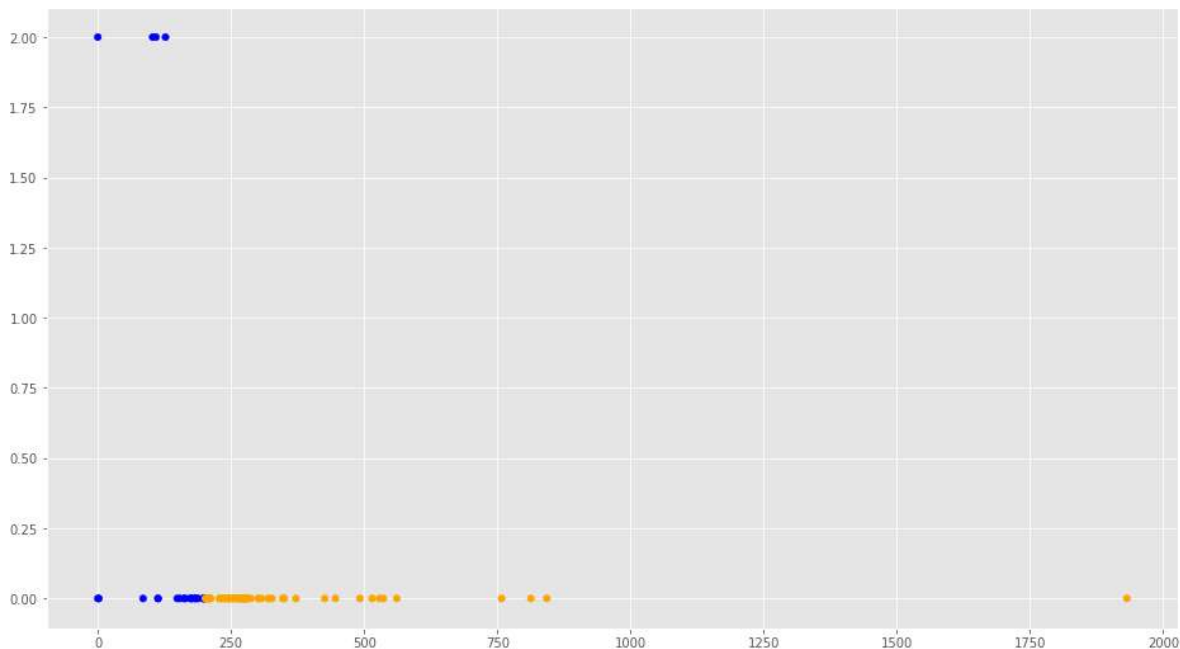
In [103]:

```
1 fb_posts.drop(['post_id', 'post_url', 'time'],1).hist()  
2 plt.show()
```



In [104]:

```
1 #vamos a Visualizar los datos de entrada
2 colores=['orange','blue']
3 tamanios=[30,60]
4
5 filtered_data = fb_posts[(fb_posts['numWord'] <= 3500) & (fb_posts['shares'] <= 80000)]
6
7 f1 = filtered_data['numWord'].values
8 f2 = filtered_data['shares'].values
9
10 asignar=[]
11 for index, row in filtered_data.iterrows():
12     if(row['numWord']>200):
13         asignar.append(colores[0])
14     else:
15         asignar.append(colores[1])
16
17 plt.scatter(f1, f2, c=asignar, s=tamanios[0])
18 plt.show()
```



In [105]:

```
1
2 # Asignamos nuestra variable de entrada X para entrenamiento y las etiquetas Y.
3 dataX =filtered_data[["numWord"]]
4 X_train = np.array(dataX)
5 y_train = filtered_data['shares'].values
6
7 # Creamos el objeto de Regresión Linear
8 regr = linear_model.LinearRegression()
9
10 # Entrenamos nuestro modelo
11 regr.fit(X_train, y_train)
12
13 # Hacemos las predicciones que en definitiva una línea (en este caso, al ser 2D)
14 y_pred = regr.predict(X_train)
15
16 # Veamos los coeficientes obtenidos, En nuestro caso, serán la Tangente
17 print('Coefficients: \n', regr.coef_)
18 # Este es el valor donde corta el eje Y (en X=0)
19 print('Independent term: \n', regr.intercept_)
20 # Error Cuadrado Medio
21 print("Mean squared error: %.2f" % mean_squared_error(y_train, y_pred))
22 # Puntaje de Varianza. El mejor puntaje es un 1.0
23 print('Variance score: %.2f' % r2_score(y_train, y_pred))
24
25
26 #Visualizamos la recta que se obtuvo
27 plt.scatter(X_train[:,0], y_train, c=asignar, s=tamamos[0])
28 plt.plot(X_train[:,0], y_pred, color='red', linewidth=3)
29
30 plt.xlabel('Cantidad de Palabras')
31 plt.ylabel('Compartido en Redes')
32 plt.title('Regresión Lineal')
33
34 plt.show()
```

Coefficients:

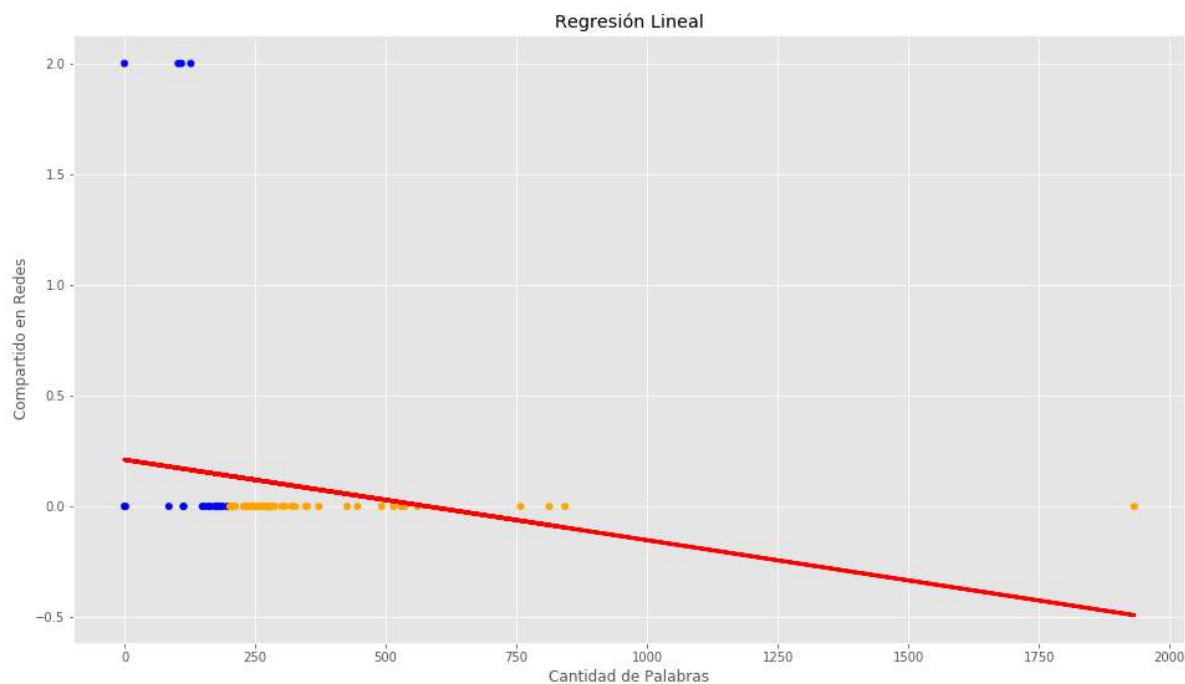
[-0.00036313]

Independent term:

0.21088541264684996

Mean squared error: 0.19

Variance score: 0.04



In [106]:

```
1 y_Dosmil = regr.predict([[2000]])  
2 print(int(y_Dosmil))
```

0

In [107]:

```
1 suma = (filtered_data["likes"] + filtered_data['comments'].fillna(0))  
2  
3 dataX2 =pd.DataFrame()  
4 dataX2["numWord"] = filtered_data["numWord"]  
5 dataX2["suma"] = suma  
6 XY_train = np.array(dataX2)  
7 z_train = filtered_data['shares'].values
```

In [108]:

```
1 # Creamos un nuevo objeto de Regresión Lineal
2 regr2 = linear_model.LinearRegression()
3
4 # Entrenamos el modelo, esta vez, con 2 dimensiones
5 # obtendremos 2 coeficientes, para graficar un plano
6 regr2.fit(XY_train, z_train)
7
8 # Hacemos la predicción con la que tendremos puntos sobre el plano hallado
9 z_pred = regr2.predict(XY_train)
10
11 # Los coeficientes
12 print('Coefficients: \n', regr2.coef_)
13 # Error cuadrático medio
14 print("Mean squared error: %.2f" % mean_squared_error(z_train, z_pred))
15 # Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
16 print('Variance score: %.2f' % r2_score(z_train, z_pred))
```

Coefficients:

[-0.0003873 0.00950019]

Mean squared error: 0.19

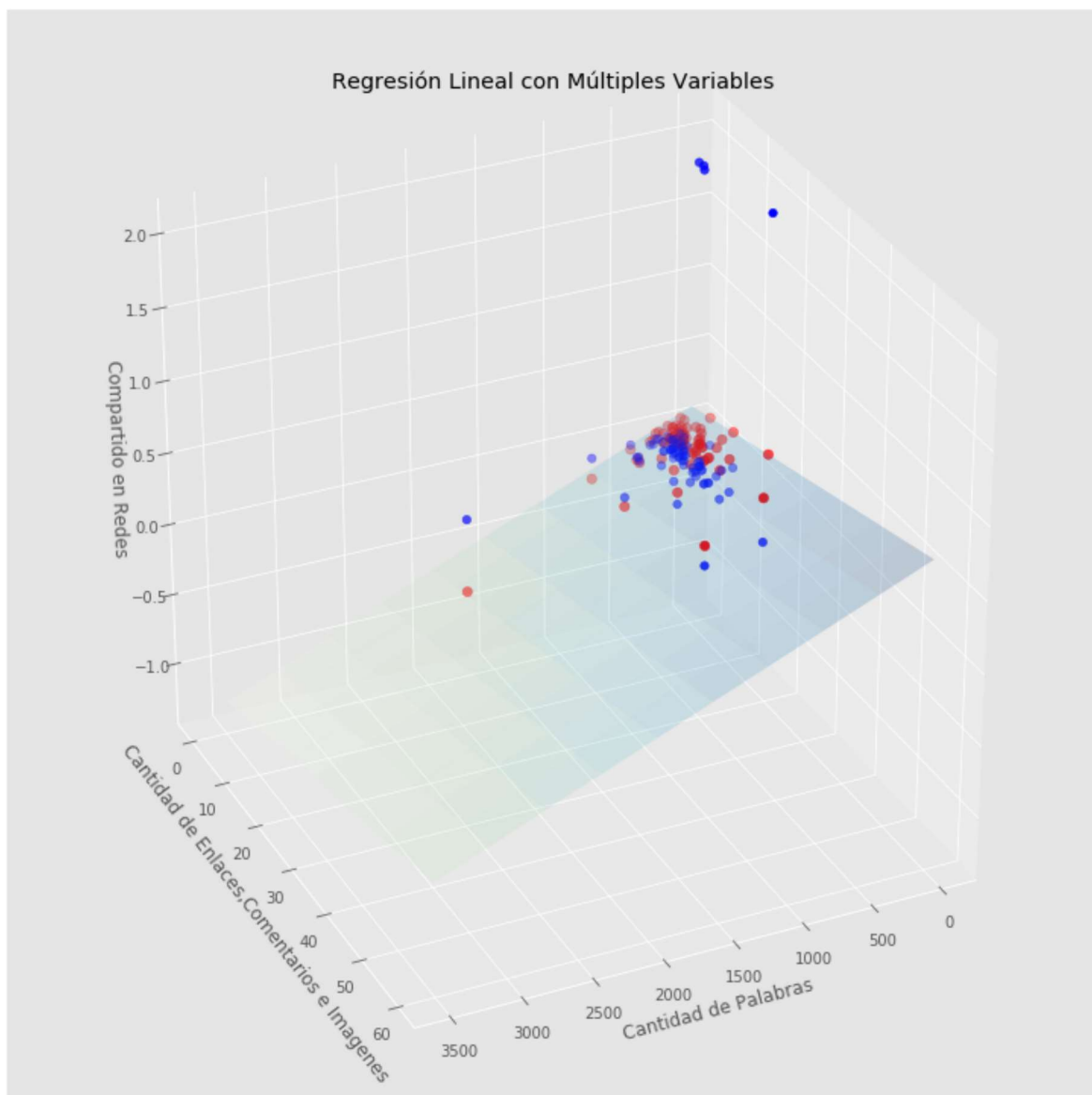
Variance score: 0.05

In [114]:

```

1 fig = plt.figure(figsize=(10,10))
2 ax = Axes3D(fig)
3 xx, yy = np.meshgrid(np.linspace(0, 3500, num=10), np.linspace(0, 60, num=10))
4 nuevoX = (regr2.coef_[0] * xx)
5 nuevoY = (regr2.coef_[1] * yy)
6 z = (nuevoX + nuevoY + regr2.intercept_)
7 ax.plot_surface(xx, yy, z, alpha=0.2, cmap='GnBu',)
8 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue',s=30)
9 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red',s=40)
10 ax.view_init(elev=30., azimuth=65)
11 ax.set_xlabel('Cantidad de Palabras')
12 ax.set_ylabel('Cantidad de Enlaces,Comentarios e Imagenes')
13 ax.set_zlabel('Compartido en Redes')
14 ax.set_title('Regresión Lineal con Múltiples Variables')
15 plt.show()

```



In [115]:

```

1 z_Dosmil = regr2.predict([[2000, 10+4+6]])
2 print(int(z_Dosmil))

```

0

In [116]:

```

1 import random
2
3 random.seed(1)
4 esteban= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
5 gloria = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
6 esteban_win_pct = 69.13
7 gloria_win_pct = 29.28
8 number_of_sims = 1000
9 total_wards = 0
10 total_esteban_votes = 0
11 total_gloria_votes = 0
12 total_votes = 0
13 for i in range(number_of_sims):
14     my_input = open('dato.txt')
15     for line in my_input:
16         total_wards += 1
17         fields = line.strip().split()
18         num_voters = int(fields[0])
19         carrasco_votes_in_ward = 0
20         lasso_votes_in_ward = 0
21         for j in range(num_voters):
22             random_num = random.random() * 100
23             if random_num <= esteban_win_pct:
24                 lasso_votes_in_ward += 1
25                 total_gloria_votes += 1
26             elif random_num <= (esteban_win_pct + gloria_win_pct) :
27                 carrasco_votes_in_ward += 1
28                 total_esteban_votes += 1
29                 total_votes += 1
30             carrasco_digit = int(str(carrasco_votes_in_ward)[0])
31             lasso_digit = int(str(lasso_votes_in_ward)[0])
32             gloria[carrasco_digit] += 1
33             lasso_digits[lasso_digit] += 1
34         if i % 100 == 0:
35             print('Simulacion con ', i)
36
37 gloria_win_pct = 100.0 * total_esteban_votes / total_votes
38 esteban_win_pct = 100.0 * total_gloria_votes / total_votes
39 print('Esteban Albornoz V gana con el:', gloria_win_pct , '%')
40 print('Gloria Astudillo Loor gana con el:', esteban_win_pct, '%')

```

```

Simulacion con 0
Simulacion con 100
Simulacion con 200
Simulacion con 300
Simulacion con 400
Simulacion con 500
Simulacion con 600
Simulacion con 700
Simulacion con 800
Simulacion con 900
Esteban Albornoz V gana con el: 28.875 %
Gloria Astudillo Loor gana con el: 69.4875 %

```

In [117]:

```

1 import simpy
2 import random
3 import matplotlib.pyplot as pp
4 import random
5
6 %matplotlib inline
7
8 MAXIMO_VOTANTES = 50
9 NUMERO_MESAS = 3
10 TIEMPO_VOTACION = 10
11 TIEMPO_LLEGADA = 5
12 TIEMPO_SIMULACION = 50
13
14 votos = {}
15
16 class Lugar_Votacion():
17
18     def __init__(self, environment, numero_mesas, tiempo_votacion):
19         self.env = environment
20         self.mesa = simpy.Resource(environment, numero_mesas)
21         self.tiempo_votacion = tiempo_votacion
22
23     def atender_votante(self, votante):
24         yield self.env.timeout(random.randint(TIEMPO_VOTACION - 5, TIEMPO_VOTACION + 5))
25
26 def llegada_votante(env, nombre, Lugar_Votacion):
27     print('El %s llega' % (nombre))
28     with Lugar_Votacion.mesa.request() as maquina:
29         yield maquina
30         print('El %s ingresa' % (nombre))
31         yield env.process(Lugar_Votacion.atender_votante(nombre))
32         print('El %s entra'%(nombre))
33         print('El %s recibe su certificado de votación.'%(nombre))
34         print('El %s sale del Recinto Electoral.'%(nombre))
35         votos[nombre] = random.randint(1, 3)
36
37
38 def simular(env, mesas, tiempo_votacion, intervalo):
39     lugar = Lugar_Votacion(env, mesas, tiempo_votacion)
40     for i in range(3):
41         env.process(llegada_votante(env, 'votante %d' % (i + 1), lugar))
42     while True:
43         yield env.timeout(random.randint(intervalo - 3, intervalo + 3))
44         i += 1
45         env.process(llegada_votante(env, 'votante %d' % (i + 1), lugar))
46
47
48 print('Lugar de votación')
49
50 env = simpy.Environment()
51 env.process(simular(env, NUMERO_MESAS, TIEMPO_VOTACION, TIEMPO_LLEGADA))
52 env.run(until=TIEMPO_SIMULACION)
53
54 print("Descripción de votos")
55 print(votos)

```

Lugar de votación
El votante 1 llega

```
El votante 2 llega
El votante 3 llega
El votante 1 ingresa
El votante 2 ingresa
El votante 3 ingresa
El votante 1 entra
El votante 1 recibe su certificado de votación.
El votante 1 sale del Recinto Electoral.
El votante 4 llega
El votante 4 ingresa
El votante 3 entra
El votante 3 recibe su certificado de votación.
El votante 3 sale del Recinto Electoral.
El votante 5 llega
El votante 2 entra
El votante 2 recibe su certificado de votación.
El votante 2 sale del Recinto Electoral.
El votante 5 ingresa
El votante 4 entra
El votante 4 recibe su certificado de votación.
El votante 4 sale del Recinto Electoral.
El votante 6 llega
El votante 6 ingresa
El votante 7 llega
El votante 7 ingresa
El votante 5 entra
El votante 5 recibe su certificado de votación.
El votante 5 sale del Recinto Electoral.
El votante 8 llega
El votante 8 ingresa
El votante 6 entra
El votante 6 recibe su certificado de votación.
El votante 6 sale del Recinto Electoral.
El votante 7 entra
El votante 7 recibe su certificado de votación.
El votante 7 sale del Recinto Electoral.
El votante 9 llega
El votante 9 ingresa
El votante 8 entra
El votante 8 recibe su certificado de votación.
El votante 8 sale del Recinto Electoral.
El votante 10 llega
El votante 10 ingresa
Descripción de votos
{'votante 1': 3, 'votante 3': 1, 'votante 2': 3, 'votante 4': 2, 'votante
5': 3, 'votante 6': 2, 'votante 7': 3, 'votante 8': 3}
```

Conclusiones

Con las regresiones se puede pronosticar datos fiables para tomar decisiones y tener posibles decisiones. Con este sistema podemos aplicar las soluciones y solucionar problemas que se pueda tener. En tiempo de elecciones se puede sacar una predicción del candidato que tiene más probabilidad de ganar las elecciones.

Referencias

- [1]: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/>

- [\(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/\)](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/)
- [2]: <https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>
(<https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>)
 - [3]: <https://eprints.ucm.es/48804/1/TFM%20Manuel%20Alejandro%20Rodriguez%20Santana.pdf>
(<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)

In []:

1	
---	--