ExamenInterciclo

December 20, 2020

1 UNIVERSIDAD POLITECNICA SALESIANA

Nombre: Jordan Murillo

Carrera: Ingenieria en Sistemas Décimo Ciclo

Materia: Simulación

Profesor: Diego Quisi

2 Examen Interciclo

[1]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
from facebook_scraper import get_posts
import numpy as np
import seaborn as sb
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

[2]:
```
#Metodo para contar las palabras del post obtenido
def contador(post):
    cont=0
    for char in post:
            cont+=1
    return cont

publicaciones = []
# Obtenemos la información de las publicaciones
for post in get_posts('LassoGuillermo', pages=10):
        post['Word count'] = len(post['text'])
        publicaciones.append(post)
facebook_posts = pd.DataFrame(publicaciones)
```

```
facebook_posts.describe()
```

[2]:

	likes	comments	shares	Word count
count	38.000000	38.000000	38.000000	38.000000
mean	1691.500000	302.131579	9.842105	218.973684
std	947.087705	282.031163	55.431734	111.274068
min	431.000000	32.000000	0.000000	64.000000
25%	1043.000000	142.750000	0.000000	118.000000
50%	1660.500000	201.500000	0.000000	206.500000
75%	2102.000000	332.000000	0.000000	277.250000
max	4933.000000	1333.000000	341.000000	518.000000

Los Post obtenidos son a nivel de candidatos presidenciales debido a que obtener datos por asambleistas del azuay son muy pocos los que se pueden obtener.

[3]:
```
#Lista donde se guardaran las publicaciones obtenidas
posts = []
#Obtenemos los post del candidato guillermo lasso
for post in get_posts('LassoGuillermo', pages=10):
    post['numWord']=contador(post['text'])
    posts.append(post)
#Obtenemos los post del candidato Paul Carrasco
for post in get_posts('ecuarauz2021', pages=10):
    post['numWord']=contador(post['text'])
    posts.append(post)
fb_posts = pd.DataFrame(posts)
```

[4]:
```
file = open("likes.txt","w")
for like in fb_posts['likes']:
    file.write(str(like)+'\n')
file.close()
```

[5]:
```
fb_posts.describe()
```

[5]:

	likes	comments	shares	numWord
count	76.000000	76.000000	76.000000	76.000000
mean	1925.052632	505.842105	196.210526	203.723684
std	1329.410826	847.911757	538.722546	132.344460
min	202.000000	6.000000	0.000000	34.000000
25%	1031.500000	121.750000	0.000000	92.500000
50%	1796.000000	196.000000	0.000000	180.000000
75%	2448.500000	437.000000	0.000000	272.750000
max	9110.000000	5116.000000	2665.000000	687.000000

[6]:
```
fb_posts.head()
```

[6]:

	post_id	text	post_text	shared_text
0	4845903942117477	El Ecuador está listo para el CAMBIO .\n\nEn…	El Ecuador está listo para el CAMBIO .\n\nEn…	
1	4845372462170625	¡Hoy celebramos la cantonización de Sozoranga!…	¡Hoy celebramos la cantonización de Sozoranga!…	
2	4844341652273706	Con la salud no se juega. Junto al Dr. Alfredo…	Con la salud no se juega. Junto al Dr. Alfredo…	
3	4841574735883731	¡GRACIAS LOJA!\n\nLa alegría por la llegada de…	¡GRACIAS LOJA!\n\nLa alegría por la llegada de…	
4	4841197019254836	Con Guido Chiriboga , tenemos la #CapacidadPar…	Con Guido Chiriboga , tenemos la #CapacidadPar…	

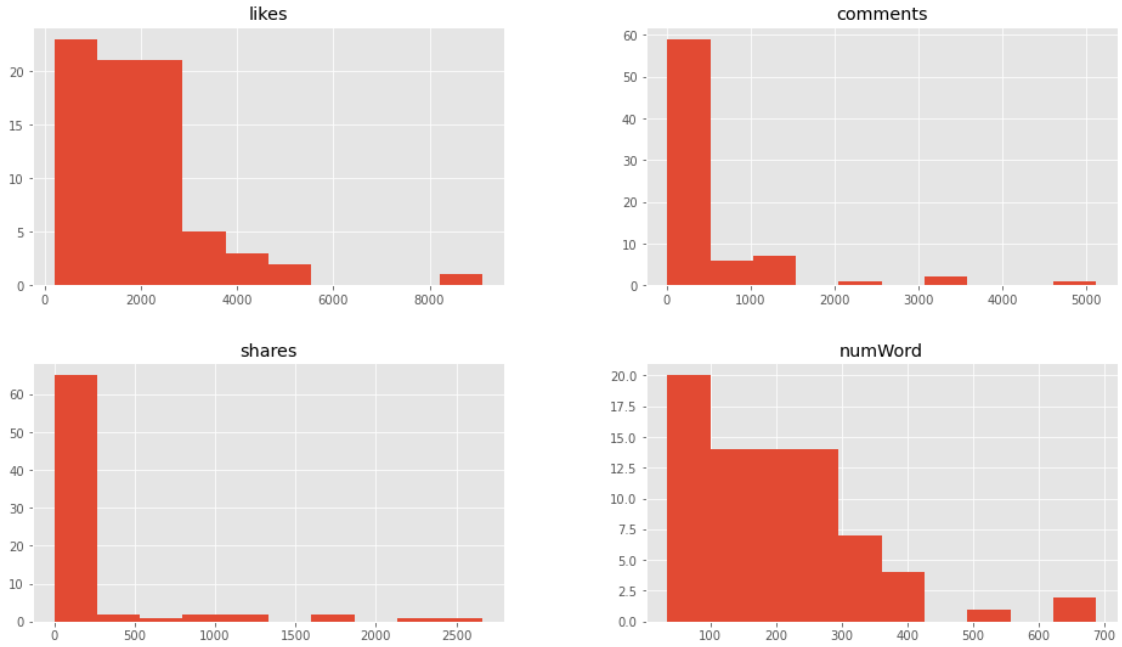
	time	image	video
0	2020-12-20 19:12:15	https://scontent.fgye1-1.fna.fbcdn.net/v/t1.0-…	None
1	2020-12-20 15:47:00	None	None
2	2020-12-20 10:38:01	None	https://video.fgye1-1.fna.fbcdn.net/v/t42.1790…
3	2020-12-19 18:45:41	https://scontent.fgye1-1.fna.fbcdn.net/v/t1.0-…	None
4	2020-12-19 16:26:19	None	https://video.fgye1-1.fna.fbcdn.net/v/t42.1790…

	video_thumbnail	video_id	likes	comments	shares	post_url	link
0	None	None	1388	122	0	https://facebook.com/story.php?story_fbid=4845…	None
1	None	None	431	52	0	https://facebook.com/story.php?story_fbid=4845…	None
2	https://scontent.fgye1-1.fna.fbcdn.net/v/t15.5…	232917304870895	1037	176	0	https://facebook.com/watch?v=232917304870895	None
3	None	None	2327	238	0	https://facebook.com/story.php?story_fbid=4841…	None
4	https://scontent.fgye1-1.fna.fbcdn.net/v/t15.5…	401847227903830	706	73	0	https://facebook.com/watch?v=401847227903830	None

	user_id	images	numWord
0	401049006603015	[https://scontent.fgye1-1.fna.fbcdn.net/v/t1.0…	190
1	401049006603015	[]	224
2	401049006603015	[]	118
3	401049006603015	[https://scontent.fgye1-1.fna.fbcdn.net/v/t1.0…	278

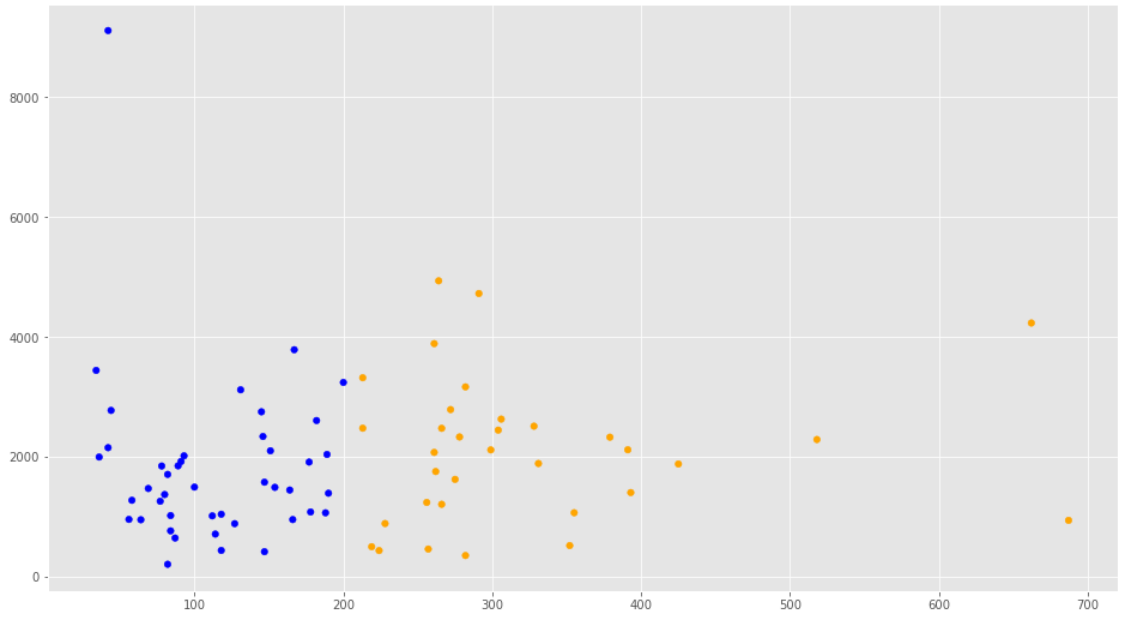
```
4   401049006603015                                          []        114
```

[7]:
```
fb_posts.drop(['post_id','post_url', 'time'],1).hist()
plt.show()
```



Visualizamos la cantidad de palabras contra likes

[8]:
```
#vamos a Visualizar los datos de entrada
colores=['orange','blue']
tamanios=[30,60]

filtered_data = fb_posts[(fb_posts['numWord'] <= 3500) & (fb_posts['likes'] <=␣
↪80000)]

f1 = filtered_data['numWord'].values
f2 = filtered_data['likes'].values

asignar=[]
for index, row in filtered_data.iterrows():
    if(row['numWord']>200):
        asignar.append(colores[0])
    else:
        asignar.append(colores[1])

plt.scatter(f1, f2, c=asignar, s=tamanios[0])
plt.show()
```

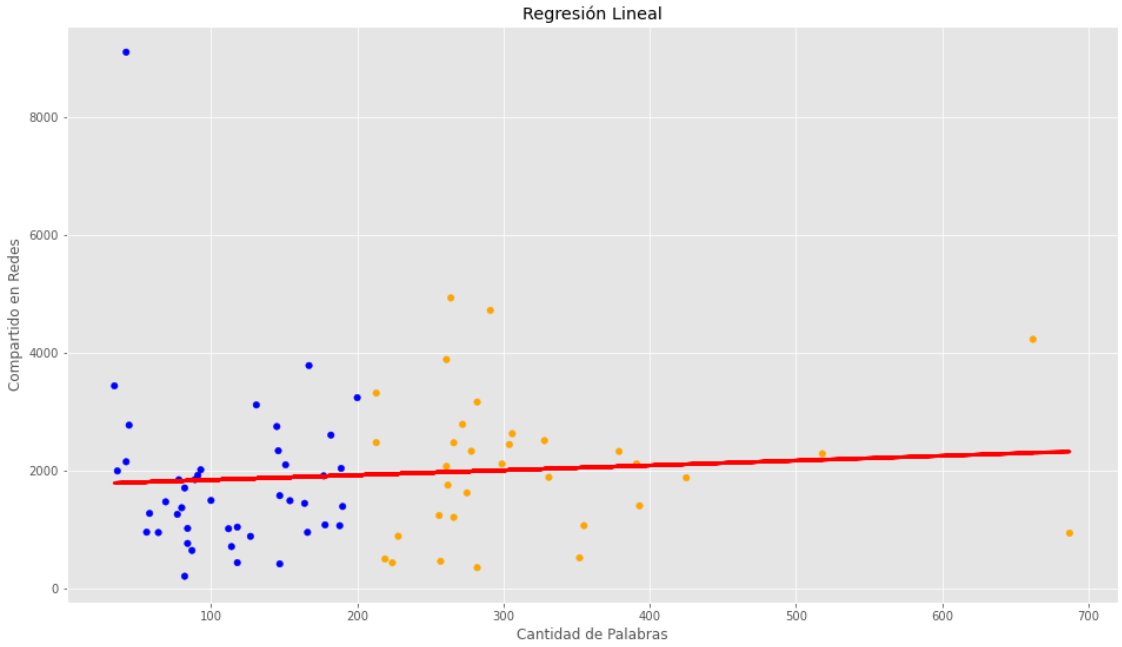


Regresión Lineal Simple

[9]:
```
# Asignamos nuestra variable de entrada X para entrenamiento y las etiquetas Y.
dataX =filtered_data[["numWord"]]
X_train = np.array(dataX)
y_train = filtered_data['likes'].values

# Creamos el objeto de Regresión Linear
regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(X_train, y_train)

# Hacemos las predicciones que en definitiva una línea (en este caso, al ser 2D)
y_pred = regr.predict(X_train)

# Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangente
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
print("Mean squared error: %.2f" % mean_squared_error(y_train, y_pred))
# Puntaje de Varianza. El mejor puntaje es un 1.0
print('Variance score: %.2f' % r2_score(y_train, y_pred))


#Visualizamos la recta que se obtuvo
```

```
plt.scatter(X_train[:,0], y_train,  c=asignar, s=tamanios[0])
plt.plot(X_train[:,0], y_pred, color='red', linewidth=3)

plt.xlabel('Cantidad de Palabras')
plt.ylabel('Compartido en Redes')
plt.title('Regresión Lineal')

plt.show()
```

```
Coefficients:
 [0.81771561]
Independent term:
 1758.4645941252807
Mean squared error: 1732521.26
Variance score: 0.01
```



Prediccion 1

[10]:
```
y_Dosmil = regr.predict([[2000]])
print(int(y_Dosmil))
```

```
3393
```

Regresion Lineal con Multiples Variables

[11]:
```
suma = (filtered_data["shares"] + filtered_data['comments'].fillna(0))
```

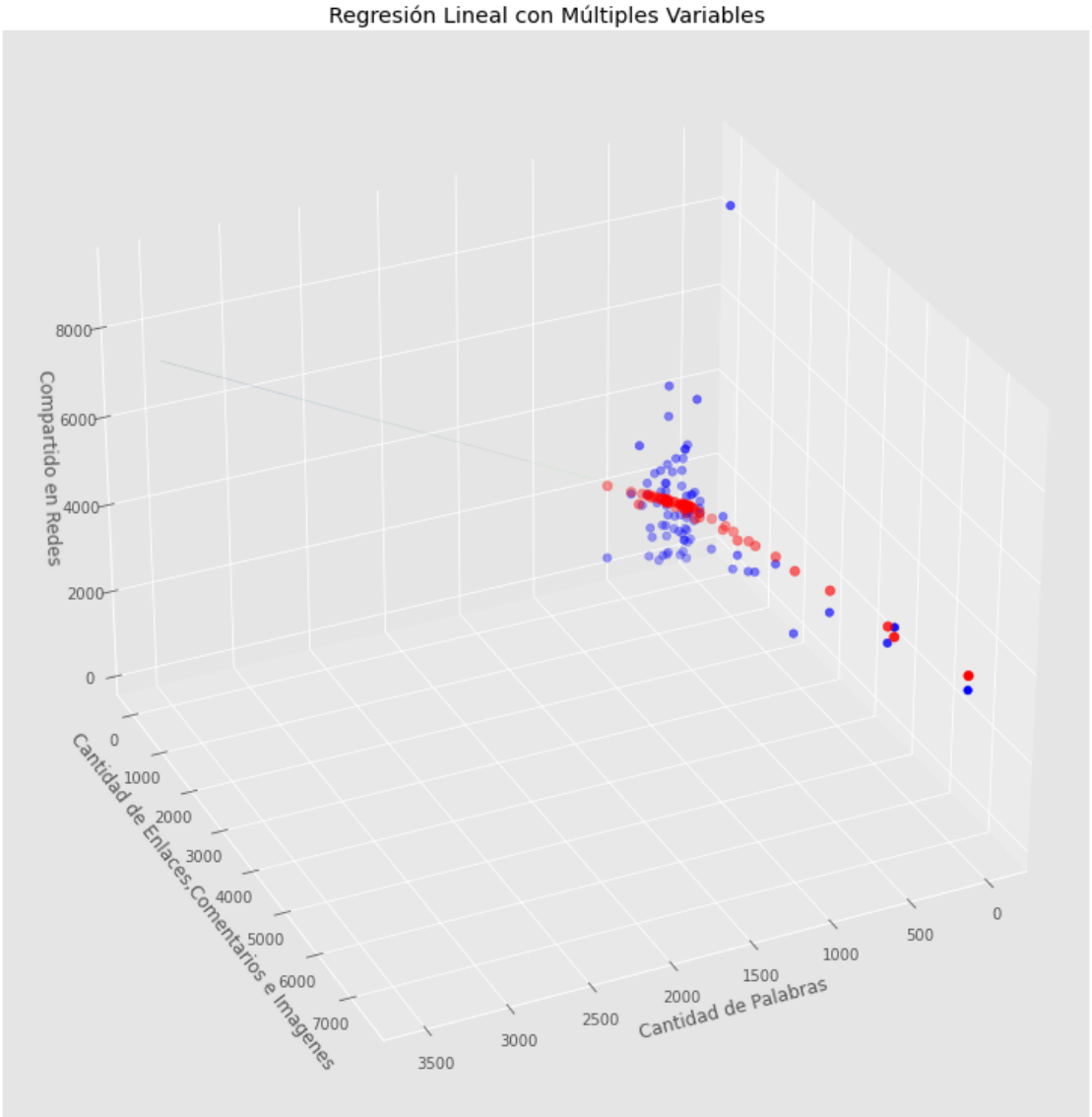
```
dataX2 =pd.DataFrame()
dataX2["numWord"] = filtered_data["numWord"]
dataX2["suma"] = suma
XY_train = np.array(dataX2)
z_train = filtered_data['likes'].values
```

[12]:
```
# Creamos un nuevo objeto de Regresión Lineal
regr2 = linear_model.LinearRegression()

# Entrenamos el modelo, esta vez, con 2 dimensiones
# obtendremos 2 coeficientes, para graficar un plano
regr2.fit(XY_train, z_train)

# Hacemos la predicción con la que tendremos puntos sobre el plano hallado
z_pred = regr2.predict(XY_train)

# Los coeficientes
print('Coefficients: \n', regr2.coef_)
# Error cuadrático medio
print("Mean squared error: %.2f" % mean_squared_error(z_train, z_pred))
# Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
print('Variance score: %.2f' % r2_score(z_train, z_pred))
```

```
Coefficients:
 [1.75995737 0.31766346]
Mean squared error: 1567232.68
Variance score: 0.10
```

[13]:
```
fig = plt.figure(figsize=(10,10))
ax = Axes3D(fig)

# Creamos una malla, sobre la cual graficaremos el plano
xx, yy = np.meshgrid(np.linspace(0, 3500, num=10), np.linspace(0, 60, num=10))

# calculamos los valores del plano para los puntos x e y
nuevoX = (regr2.coef_[0] * xx)
nuevoY = (regr2.coef_[1] * yy)

# calculamos los correspondientes valores para z. Debemos sumar el punto de␣
 ↪intercepción
z = (nuevoX + nuevoY + regr2.intercept_)

# Graficamos el plano

ax.plot_surface(xx, yy, z, alpha=0.2, cmap='GnBu',)
```

```
# Graficamos en azul los puntos en 3D
ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue',s=30)

# Graficamos en rojo, los puntos que
ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red',s=40)

# con esto situamos la "camara" con la que visualizamos
ax.view_init(elev=30., azim=65)

ax.set_xlabel('Cantidad de Palabras')
ax.set_ylabel('Cantidad de Enlaces,Comentarios e Imagenes')
ax.set_zlabel('Compartido en Redes')
ax.set_title('Regresión Lineal con Múltiples Variables')
plt.show()
```



Prediccion 2

[14]:
```
# Si quiero predecir cuántos "Shares" voy a obtener por un artículo con:
# 2000 palabras y con enlaces: 10, comentarios: 4, imagenes: 6
# según nuestro modelo, hacemos:

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

```
4869
```

Simulacion tomando en cuenta el numero de likes

[15]:
```
import random

random.seed(1)
lasso_digits = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
arauz_digits = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
lasso_win_pct = 69.13
arauz_win_pct = 29.28
number_of_sims = 1000
total_wards = 0
total_arauz_votes = 0
total_lasso_votes = 0
total_votes = 0
for i in range(number_of_sims):
    my_input = open('likes.txt')
    for line in my_input:
        total_wards += 1
        fields = line.strip().split()
        num_voters = int(fields[0])
        arauz_votes_in_ward = 0
        lasso_votes_in_ward = 0
        for j in range(num_voters):
            random_num = random.random() * 100
            if random_num <= lasso_win_pct:
                lasso_votes_in_ward += 1
                total_lasso_votes += 1
            elif random_num <= (lasso_win_pct + arauz_win_pct) :
                arauz_votes_in_ward += 1
                total_arauz_votes += 1
            total_votes += 1
        arauz_digit = int(str(arauz_votes_in_ward)[0])
        lasso_digit = int(str(lasso_votes_in_ward)[0])
        arauz_digits[arauz_digit] += 1
        lasso_digits[lasso_digit] += 1
```

```
    if i % 100 == 0:
        print('Simulacion con ', i)

arauz_win_pct = 100.0 * total_arauz_votes / total_votes
lasso_win_pct = 100.0 * total_lasso_votes / total_votes
print('GUILLERMO LASSO gana con el:', lasso_win_pct , '%')
print('ARAUZ gana con el:', arauz_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
GUILLERMO LASSO gana con el: 69.13018646106737 %
ARAUZ gana con el: 29.280210383858268 %
```

[1]:
```
import simpy
import random
import matplotlib.pyplot as pp
import random

%matplotlib inline

MAXIMO_VOTANTES = 50
NUMERO_MESAS = 3
TIEMPO_VOTACION = 10
TIEMPO_LLEGADA = 5
TIEMPO_SIMULACION = 50

# Creamos un diccionario para almacenar las horas en que se sufragan los␣
 ↪votantes..
votos = {}

class Lugar_Votacion():

    def __init__(self, environment, numero_mesas, tiempo_votacion):
        # Guardar el entorno en una variable
        self.env = environment
        # Recurso que representa las mesas electorales
        self.mesa = simpy.Resource(environment, numero_mesas)
        # Variable para el tiempo de atención.
        self.tiempo_votacion = tiempo_votacion
```

```
    def atender_votante(self, votante):
        # Simular la votación
        yield self.env.timeout(random.randint(TIEMPO_VOTACION - 5,␣
↪TIEMPO_VOTACION + 5))

def llegada_votante(env, nombre, Lugar_Votacion):
    print('El %s llega a sufragar' % (nombre))

    # Especificamos que vamos a usar un recurso (Resource) que representa
    with Lugar_Votacion.mesa.request() as maquina:
        # Ocupamos la mesa electoral.
        yield maquina
        # El votante ingresa a votar.
        print('El %s ingresa a sufragar:' % (nombre))
        # Procesamos la operacion de sufragio
        yield env.process(Lugar_Votacion.atender_votante(nombre))
        # Una vez que termina la llamada con 'yield', se indica que se ha␣
↪atendido al votante.
        print('EL %s entra a sufragar'%(nombre))
        print('El %s recibe su certificado de votación.'%(nombre))
        print('El %s sale del Recinto Electoral.'%(nombre))
        votos[nombre] = random.randint(1, 3) #Aleatorio para los votos a las␣
↪diferentes listas.


def simular(env, mesas, tiempo_votacion, intervalo):
    lugar = Lugar_Votacion(env, mesas, tiempo_votacion)
    # Iniciamos con 3 votantes
    for i in range(3):
        env.process(llegada_votante(env, 'votante-%d' % (i + 1), lugar))

    # Ejecutamos la simulación
    while True:
        yield env.timeout(random.randint(intervalo - 3, intervalo + 3))
        i += 1
        # generamos mas votantes
        env.process(llegada_votante(env, 'votante-%d' % (i + 1), lugar))


print('Recinto')

# Creamos el entorno de simulacion
env = simpy.Environment()
env.process(simular(env, NUMERO_MESAS, TIEMPO_VOTACION, TIEMPO_LLEGADA))

# Ejecutamos el proceso durante el tiempo de simulacion
```

```
env.run(until=TIEMPO_SIMULACION)

print("Detalle de los votos")
print(votos)
```

```
Recinto
El votante-1 llega a sufragar
El votante-2 llega a sufragar
El votante-3 llega a sufragar
El votante-1 ingresa a sufragar:
El votante-2 ingresa a sufragar:
El votante-3 ingresa a sufragar:
El votante-4 llega a sufragar
EL votante-3 entra a sufragar
El votante-3 recibe su certificado de votación.
El votante-3 sale del Recinto Electoral.
El votante-4 ingresa a sufragar:
El votante-5 llega a sufragar
EL votante-1 entra a sufragar
El votante-1 recibe su certificado de votación.
El votante-1 sale del Recinto Electoral.
El votante-5 ingresa a sufragar:
EL votante-2 entra a sufragar
El votante-2 recibe su certificado de votación.
El votante-2 sale del Recinto Electoral.
El votante-6 llega a sufragar
El votante-6 ingresa a sufragar:
EL votante-4 entra a sufragar
El votante-4 recibe su certificado de votación.
El votante-4 sale del Recinto Electoral.
EL votante-5 entra a sufragar
El votante-5 recibe su certificado de votación.
El votante-5 sale del Recinto Electoral.
El votante-7 llega a sufragar
El votante-7 ingresa a sufragar:
El votante-8 llega a sufragar
El votante-8 ingresa a sufragar:
El votante-9 llega a sufragar
EL votante-6 entra a sufragar
El votante-6 recibe su certificado de votación.
El votante-6 sale del Recinto Electoral.
EL votante-8 entra a sufragar
El votante-8 recibe su certificado de votación.
El votante-8 sale del Recinto Electoral.
El votante-9 ingresa a sufragar:
El votante-10 llega a sufragar
El votante-10 ingresa a sufragar:
```

```
EL votante-9 entra a sufragar
El votante-9 recibe su certificado de votación.
El votante-9 sale del Recinto Electoral.
EL votante-7 entra a sufragar
El votante-7 recibe su certificado de votación.
El votante-7 sale del Recinto Electoral.
El votante-11 llega a sufragar
El votante-11 ingresa a sufragar:
El votante-12 llega a sufragar
El votante-12 ingresa a sufragar:
EL votante-10 entra a sufragar
El votante-10 recibe su certificado de votación.
El votante-10 sale del Recinto Electoral.
El votante-13 llega a sufragar
El votante-13 ingresa a sufragar:
EL votante-11 entra a sufragar
El votante-11 recibe su certificado de votación.
El votante-11 sale del Recinto Electoral.
Detalle de los votos
{'votante-3': 2, 'votante-1': 1, 'votante-2': 2, 'votante-4': 1, 'votante-5': 3,
'votante-6': 3, 'votante-8': 1, 'votante-9': 1, 'votante-7': 1, 'votante-10': 1,
'votante-11': 3}
```

3 Conclusión

Gracias a la ayuda de las regresiones se puede tener datos pronosticados fiables para tomar decisiones y conocer posibles resultados. Este tipo de sistemas ayudan muchos para aplicar soluciones a problemas que se puedan presentar. Sirve de mucho además, porque en tiempo de elecciones las población siempre esta al pendiente de quién será el próximo candidato ganador.

4 Referencias

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/
- https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/
- https://eprints.ucm.es/48804/1/TFM%20Manuel%20Alejandro%20Rodriguez%20Santana.pdf
- http://opac.pucv.cl/pucv_txt/txt-8000/UCC8094_01.pdf