

```
#pip install tweepy
import tweepy
from datetime import datetime, date, time, timedelta
import calendar
import pandas as pd
import numpy as np
import seaborn as sb
from matplotlib import cm
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
import json
import os
from google.colab import drive
import matplotlib.pyplot as plt
%matplotlib inline
drive.mount('/content/drive/')
```

Mounted at /content/drive/

```
consumer_key = "wJrU63rYNS51kIMsQFkbl3Ou"
consumer_secret = "pQt1yClOnRomMvUn1LGb9poDUZXA1JE3dSLrKjUAmBRakxvc2Q"
access_token = "250372770-hEGkVutoyVzn4f0kmahzNNiVqcxCY17eCiBXGvtc"
access_token_secret = "YSlFipxdcEoxWHHrrgebQqa0iAb3jFODjliNXg2nvuXeZ"
```

▼ valores del token a setear

```
#SETEAO DE API CON LOS VALORES
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
```

```
#Se muestra las publicaciones del USUARIO
cont = 0
usuario = 'ecuarauz'
tw = api.user_timeline(usuario, tweet_mode="extended", count='200')
for tweet in tw:
    cont += 1
    a = tweet.full_text.split()
    fe = datetime.today() - tweet.created_at
    print("CONTENIDO...", tweet.full_text, "https://twitter.com/" + usuario + "/status/" + str(tweet.id))

print("*****")
```

CONTENIDO... @diegodeguez que descanses en la paz de la constancia y de la sabidur

CONTENIDO... RT @pierinaescorrea: [Boletín]

Comisión de Protección Integral a Niñas, Niños y Adolescentes aprobó plan de tr

CONTENIDO... RT @TaxJusticeNet: We're delighted to announce the programme for Tax

CONTENIDO... RT @LosTiemposBol: Informe señala que Ecuador prestó a Bolivia 5.500

CONTENIDO... RT @avanoliok: Final a la causa más emblemática del "lawfare". Penos

CONTENIDO... RT @LulaOficial: O povo está cansado. Liga a televisão, tá lá o Bolso

CONTENIDO... Indéniable.
cc @BrunoLeMaire <https://t.co/4RPCA9KqVG> <https://t.co/yLKfFS2R2A> <https://twitter.c>

CONTENIDO... 🕒 <https://t.co/YoZnFQAQQS> <https://twitter.com/ecuarauz/status/140276>

CONTENIDO... @jp_koning Ecuador: Nov 2011 <https://t.co/9Pt9Rpe76H> <https://twitter.>

CONTENIDO... RT @LuisaGonzalezEc: ¿Estamos entendiendo lo qué pasa? Don V.I.P @die

CONTENIDO... RT @FernandoCedeR: ¡¡#Perenco está radicada en bsBahamasbs!!

Esto quiere decir que:

✓ Perenco no puede acogerse al Tratado Bilateral de... <https://twitter.com/ecuarauz>

CONTENIDO... 🌐 <https://twitter.com/ecuarauz/status/1402406542025109507> tamaño

CONTENIDO... Mi solidaridad con Natalie Edwards <https://twitter.com/ecuarauz/statu>

CONTENIDO... RT @MashiRafael: Me alegro que ahora ya lo sepa, presidente Bukele.
Solo en un día de octubre de 2019 el Gobierno de Moreno lanzó más bomba... <https://t>

CONTENIDO... RT @wikileaks: Unless Biden drops the #Assange case entirely and canc

CONTENIDO... ✍ <https://twitter.com/ecuarauz/status/1401384958464249858> tamaño..

CONTENIDO... 👍 <https://t.co/0XAcDIzuZh> <https://twitter.com/ecuarauz/status/140089>

CONTENIDO... RT @VickyDesintonio: El fortalecimiento y la unidad de @CompromisoRC

CONTENIDO... RT @PaolaCabezasC: ¿Decidir entre comprar la comida o toallas sanitari

CONTENIDO... RT @RafaelCompte: El proyecto de Ley Orgánica de Libre Expresión y Co

CONTENIDO... RT @OpenlabEc: Disfruta de las 18 charlas y talleres impartidos en el

Están... <https://twitter.com/ecuarauz/status/1397925905243144197> tamaño.... 21 C

CONTENIDO... RT @DesbordeEc: El #ecuador recibirá aproximadamente USD 1000 millone

CONTENIDO... Excelente informe de @eurodad y @danielmunevar sobre la deuda externa

CONTENIDO... Los cargos ad honorem sí están normados y regulados en el servicio ex

```
list_conten = []
list_urls = []
list_palabras = []
list_favoritos = []
list_dias = []
list_retweet = []
cont = 0
```

```

cont = 0
for tweet in tweepy.Cursor(api.search,q='ecuarauz', lang="es",tweet_mode="extended").items
#for tweet in api.search(q="guillermo lasso", tweet_mode="extended", count="115"):
    if 'retweeted_status' in tweet._json:
        cont += 1
        list_urls.append("https://twitter.com/user/status/"+str(tweet.id))

        list_conten.append(tweet.retweeted_status.full_text)
        list_palabras.append(len(a) - 1)
        a = tweet.retweeted_status.full_text.split()
        f = datetime.today() - tweet.created_at
        list_dias.append(f.days)
        list_retweet.append(tweet.retweet_count)
        list_favoritos.append(tweet.favorite_count)
        print("----- si")
    else:
        cont += 1
        list_conten.append(tweet.full_text)
        list_urls.append("https://twitter.com/user/status/"+str(tweet.id))
        a = tweet.full_text.split()
        list_palabras.append(len(a) - 1)
        list_favoritos.append(tweet.favorite_count)
        f = datetime.today() - tweet.created_at
        list_dias.append(f.days)

        list_retweet.append(tweet.retweet_count)
        print("-----no")

print(cont)

```

```

----- si
-----no
----- si
-----no
----- si
----- si
-----no
----- si
----- si
----- si
----- si
----- si
----- si
----- si
----- si
----- si
----- si
----- si
----- si
-----no
----- si
----- si
----- si
----- si
----- si
----- si
-----no

```

```
----- si
----- si
----- si
-----no
----- si
----- si
----- si
----- si
----- si
----- si
-----no
-----no
-----no
----- si
----- si
----- si
----- si
----- si
----- si
-----no
----- si
----- si
----- si
-----no
----- si
----- si
----- si
----- si
-----no
----- si
----- si
----- si
```

#Se visualiza la el dataframe

```
examen = {'contenido':list_conten, 'url':list_urls, 'Numero de palabras':list_palabras, 'N
```

```
df = pd.DataFrame(data = examen)
```

```
df
```

	contenido	url	Numero de palabras	Numero de favoritos
0	Formemos un grupo compacto de tuiteros Correis...	https://twitter.com/user/status/14045285939805...	37	0
1	@DianaSalazarM2 Que pasó con la información qu...	https://twitter.com/user/status/14045283061967...	18	0
2	Con @ecuarauz ,que comparte un modelo económic...	https://twitter.com/user/status/14045281806486...	18	0
3	@rcpasajefans @MashiRafael @ecuarauz Te sigo👍EC	https://twitter.com/user/status/14045269853558...	4	0
4	Formemos un grupo compacto de tuiteros Correis...	https://twitter.com/user/status/14045268103402...	4	0
...

@GinoMayCry
@GinoMayCry

```
# ll
# lis = []
# for i in df['Numero de veces retweetiadas']:
#     lis.append(i)
# print(max(ll))
```

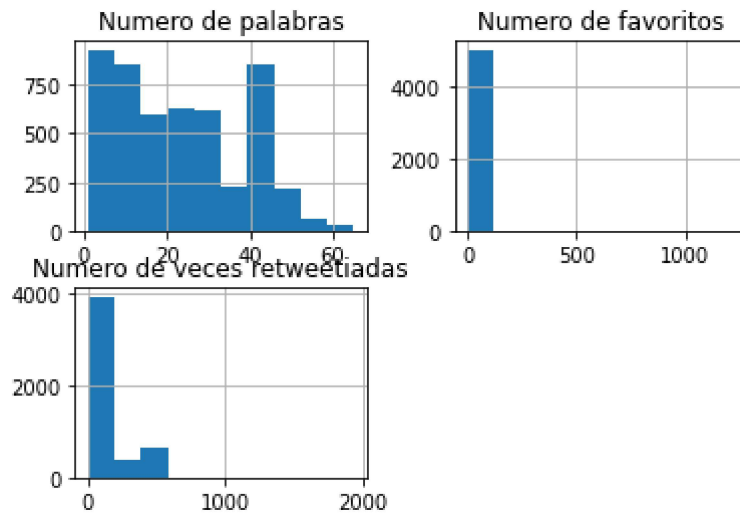
BS!!\n\nE...

```
df.describe()
```

	Numero de palabras	Numero de favoritos	Numero de dias transcurridos	Numero de veces retweetiadas
count	5000.00000	5000.00000	5000.00000	5000.00000
mean	23.15580	1.593000	3.155800	103.42440
std	14.75551	24.435719	1.634889	149.89497
min	1.00000	0.000000	0.000000	0.00000
25%	10.00000	0.000000	2.000000	0.00000
50%	21.00000	0.000000	3.000000	21.00000
75%	36.00000	0.000000	5.000000	138.00000
max	65.00000	1220.00000	5.000000	1935.00000

```
# Visualizamos rápidamente las características de entrada
```

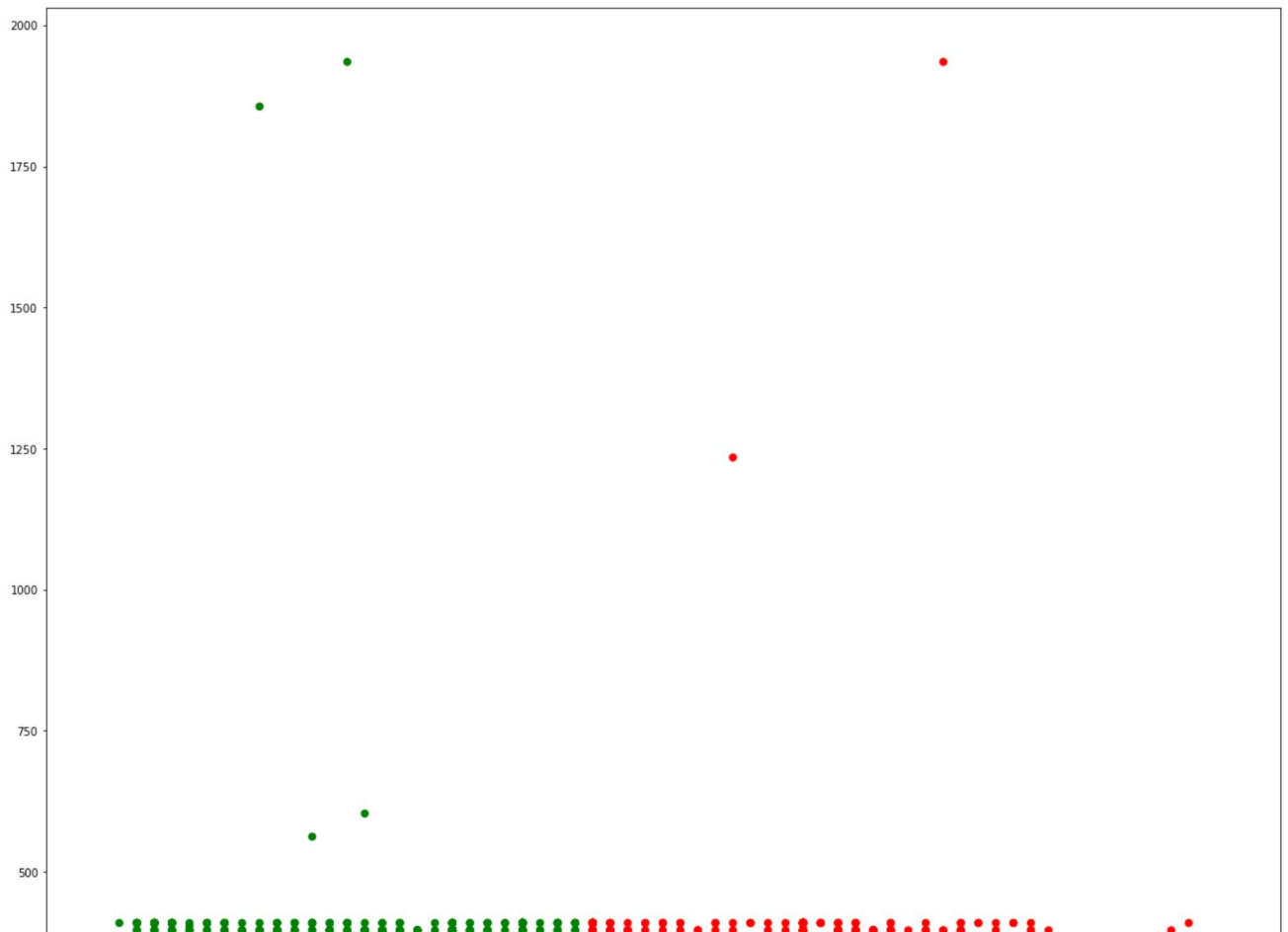
```
df.drop(['Numero de dias transcurridos','url','contenido'],1).hist()
plt.show()
```



```
#vamos a Visualizar los datos de entrada
colores=['red','green']
tamanios=[40,80]
# Vamos a RECORTAR los datos en la zona donde se concentran más los puntos
# esto es en el eje X: entre 0 y 3.000
# y en el eje Y: entre 0 y 80.000
filtered_data = df[(df['Numero de palabras'] <= 3000) & (df['Numero de veces retweetiadas']
f1 = filtered_data['Numero de palabras'].values
f2 = filtered_data['Numero de veces retweetiadas'].values

# Pintar en colores los puntos por debajo y por encima de la media de Cantidad de Palabras
asignar=[]
for index, row in filtered_data.iterrows():
    if(row['Numero de palabras']>28):
        asignar.append(colores[0])
    else:
        asignar.append(colores[1])

plt.scatter(f1, f2, c=asignar, s=tamanios[0])
plt.gcf().set_size_inches(20, 20)
plt.show()
```



Asignamos nuestra variable de entrada X para entrenamiento y las etiquetas Y.

```
dataX =filtered_data[["Numero de palabras"]]
```

```
X_train = np.array(dataX)
```

```
y_train = filtered_data['Numero de veces retweetiadas'].values
```

Creamos el objeto de Regresión Linear

```
modelo_regresion = linear_model.LinearRegression()
```

Se entrena la red

```
modelo_regresion.fit(X_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Hacemos las predicciones que en definitiva una línea (en este caso, al ser 2D)

```
y_pred = modelo_regresion.predict(X_train)
```

```
print('Coefficients: \n', modelo_regresion.coef_)
```

```
print('Independent term: \n', modelo_regresion.intercept_)
```

```
print("Cuadrado medio 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))
```

Coefficients:

```
[1.1495375]  
Independent term:  
76.80593946474751  
Cuadrado medio error: 22176.36  
Variance score: 0.01
```

