

Universidad Politecnica Salesiana

Crawler Twitter

Nombre: Javier Vazquez

Materia: Simulacion

```
In [11]: from bs4 import BeautifulSoup as bs4
import requests
from unidecode import unidecode
import re
import time
import pandas as pd
```

Selenium

- Instalaciones herramienta Selenium y el navegador Firefox
- Login en la pagina de Tweeter para que no nos bloquee el acceso y podamos extraer los tweets de una persona(credenciales unicamente creadas para pruebas)

```
In [12]: from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager

#from selenium.webdriver.chrome.options import Options
from selenium.webdriver.firefox.options import Options
from selenium.webdriver.common.keys import Keys

class Twitter:

    def __init__(self, username, password):

        self.username=username
        self.password= password
        self.browser = self.login_twitter(username, password)
        self.comentarios=[]
        self.reacciones=[]
        self.compartidos=[]
        self.contenido=[]
        self.nombre=[]
        self.username=[]

    def login_twitter(self, username, password):
        #chrome_options = Options()
        #chrome_options.add_argument("--headless")
        firefox_options= Options()
        firefox_options.add_argument("--headless")
        driver = webdriver.Firefox(executable_path="C:/Users/vazqu/Downloads/geckodriver-v0.29.1-win64/geckodriver.exe")
        driver.get("https://twitter.com/login")
        user=driver.find_element_by_name("session[username_or_email]")
        pas=driver.find_element_by_name("session[password]")
        user.send_keys(username)
        pas.send_keys(password)
        pas.send_keys(Keys.ENTER)
        time.sleep(4)
        return driver

    def get_post(self, url):
        self.browser.get(url)
        comentarios=[]
        reacciones=[]
        compartidos=[]
        contenido=[]
        self.get_vectores()
        for i in range(30):
            self.browser.execute_script("window.scrollTo(0,document.body.scrollHeight)")
            time.sleep(4)
            tweet_divs = self.browser.find_elements_by_xpath("//div[@data-testid='tweet']")
            self.get_vectores()

            print(i)

            time.sleep(3)
            html = self.browser.find_element_by_tag_name('html')
            html.send_keys(Keys.HOME)

            df= pd.DataFrame({'nombre':self.nombre,'username':self.username,'contenido':self.contenido,'comentarios':self.comentarios})
            print(df)
            df.to_csv('post_Guillermo.csv', index=False)

    def get_vectores(self):
        tweet_divs = self.browser.find_elements_by_xpath("//div[@data-testid='tweet']")
        for div in tweet_divs:
            spans = div.find_elements_by_xpath("./div/span")
            tweets = ''.join([span.text for span in spans])
            split=tweets.split('|')
            self.comentarios.append(split[-3])
            self.compartidos.append(split[-2])
            self.reacciones.append(split[-1])
            self.nombre.append(split[0])
            self.username.append(split[1])
            split.remove(split[-1])
            split.remove(split[-1])
            split.remove(split[-1])
            split.remove(split[0])
            split.remove(split[0])
            c=''.join([i for i in split])
            self.contenido.append(''.join([i for i in split]))
```

- La primera vez que realizamos un crwaler de la pagina solamente nos obtiene los primeros 5-10 tweets, implementanto un scroll podemos obtener los siguientes tweets a su vez esperando un tiempo de 4 segundos para poder obtener estos nuevos tweets
- Mediante el uso de expresiones regulares que nos facilita la herramienta podemos encontrar la etiqueta 'div' donde se encuentra cada uno de los tweets
- Se obtuvo todo ese texto dentro de la etiqueta y mediante un analisis y separadores(split) se pudo ir clasificando los diferentes contenidos, trabajando con variables globales que iran guardando concatenadamente la posicion a la que pertenece y sus valores
- Por ultimo lo guardamos dentro de archivos .csv que posteriormente se utilizara para el modelo de regresion

```
In [13]: from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as ec

if __name__ == "__main__":
    twitter = Twitter('JavierV72565554','marytigrearias99')
    twitter.get_post('https://twitter.com/LassoGuillermo')
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

           nombre      username \
0      Guillermo Lasso  @LassoGuillermo
1      Guillermo Lasso  @LassoGuillermo
2      Guillermo Lasso  @LassoGuillermo
3  Ministerio de Educación del Ecuador  @Educacion_Ec
4      Guillermo Lasso  @LassoGuillermo
..      ...
303     Guillermo Lasso  @LassoGuillermo
304     Guillermo Lasso  @LassoGuillermo
305     Guillermo Lasso  @LassoGuillermo
306     Guillermo Lasso  @LassoGuillermo
307     Guillermo Lasso  @LassoGuillermo

           contenido  comentarios  reacciones \
0  ·Primeros pasos del #PlanVacunación9100. Es un...      57      731
1  ·El #PlanVacunación9100 es \n\nSALUD\nnBIENESTA...      61      920
2  ·¡Sé parte del #PlanVacunación9100! Acudamos d...      49      756
3  ·#JuntosLoLogramos En el #DiaDelEstudiante, ...      49      438
4  ·¡Felicidades @MorejonGlenda! Ella y más depor...      98     3.8K
..      ...
303     Tendremos Centros de Reparación Integral para...      13      283
304  ·Fortaleceremos los espacios para las mujeres ...      10      276
305  ·Promoveremos la conciliación de la vida famil...      41      438
306  ·Implementaremos mecanismos de protección, den...      38      441
307  ·Encontrémonos para luchar contra la violencia...      48      515

           compartido
0              146
1              199
2              266
3              115
4              620
..      ...
303              68
304              61
305             124
306             127
307             161

[308 rows x 6 columns]
```

In [] :

Universidad Politecnica Salesiana

Modelo Regresion

Nombre: Javier Vazquez

Materia: Simulacion

```
In [1]: import pandas as pd
```

Carga de datos del archivo 'post_Guillermo.csv'

- Al momento de realizar el crawler, existen tweets que no cuentan con mucho tiempo de haber sido publicados por ende no cuentan con reacciones, comentarios o compartidos por lo que llenamos esos valores con '0'

```
In [2]: df = pd.read_csv('post_Guillermo.csv')
df.fillna(0)
df
```

	nombre	username	contenido	comentarios	reacciones	compartido
0	Guillermo Lasso	@LassoGuillermo	·Primeros pasos del #PlanVacunación9100. Es un...	57	731	146
1	Guillermo Lasso	@LassoGuillermo	·El #PlanVacunación9100 es \n\nSALUD\n\nBIENESTA...	61	920	199
2	Guillermo Lasso	@LassoGuillermo	·¡Sé parte del #PlanVacunación9100! Acudamos d...	49	756	266
3	Ministerio de Educación del Ecuador	@Educacion_Ec	·#JuntosLoLogramos En el #DíaDelEstudiante, ...	49	438	115
4	Guillermo Lasso	@LassoGuillermo	·¡Felicidades @MorejonGlenda! Ella y más depor...	98	3.8K	620
...
303	Guillermo Lasso	@LassoGuillermo	·Tendremos Centros de Reparación Integral para...	13	283	68
304	Guillermo Lasso	@LassoGuillermo	·Fortaleceremos los espacios para las mujeres ...	10	276	61
305	Guillermo Lasso	@LassoGuillermo	·Promoveremos la conciliación de la vida famil...	41	438	124
306	Guillermo Lasso	@LassoGuillermo	·Implementaremos mecanismos de protección, den...	38	441	127
307	Guillermo Lasso	@LassoGuillermo	·Encontrémonos para luchar contra la violencia...	48	515	161

308 rows × 6 columns

- En cualquiera de las redes sociales siempre existen abreviaturas para expresar la cantidad de reacciones, comentarios o compartidos, como 2000 comentarios abreviando a 2k por lo que este formato nos imposibilita trabajar con el modelo de regresión, por lo que se procede a cambiar estos valores mediante una tanto para cualquier columna del dataframe que necesite.

```
In [3]: def convert_str_to_number(x):
total_stars = 0
num_map = {'K':1000, 'M':1000000, 'B':1000000000}

if not x.isdigit():
    if len(x) > 1:
        total_stars = float(x[:-1]) * num_map.get(x[-1].upper(), 1)
    else:
        total_stars = int(x)
return int(total_stars)
```

Procesamiento

- Para obtener cuantos hastag existe dentro de un post, se comparo mediante la libreria 're' podemos contabilizar la cantidad de hastag dentro del post
- Contabilizar el numero de palabras se hizo uso de un split mediante separador de espacio
- Finalmente se cambia los formatos de la cantidad de comentarios, reacciones y compartidos de un tweet

```
In [4]: hastag=[]
numero_palabras=[]
cont=0
for i in df['contenido']:
    cont+=1
    if i==i:
        #print('pasa', cont)
        hastag.append(0)
        numero_palabras.append(0)
    else:
        cont_hastag=0
        for character in i:
            #print(character)
            if character=='#':
                cont_hastag+=1
                hastag.append(cont_hastag)
                #print('bueno')
                numero_palabras.append(len(i.split()))

df['hastag']=hastag
df['num_palabras']=numero_palabras

# convertir formato 2.3k a 2300

df['reacciones']=[convert_str_to_number(i) for i in df['reacciones']]
df['compartidos']=[convert_str_to_number(i) for i in df['compartido']]
df['comentarios']=[convert_str_to_number(i) for i in df['comentarios']]
df
```

```
Out [4]:
```

	nombre	username	contenido	comentarios	reacciones	compartido	hastag	num_palabras
0	Guillermo Lasso	@LassoGuillermo	·Primeros pasos del #PlanVacunación9100. Es un...	57	731	146	2	26
1	Guillermo Lasso	@LassoGuillermo	·El #PlanVacunación9100 es \n\nSALUD\n\nBIENESTA...	61	920	199	2	17
2	Guillermo Lasso	@LassoGuillermo	·¡Sé parte del #PlanVacunación9100! Acudamos d...	49	756	266	2	33
3	Ministerio de Educación del Ecuador	@Educacion_Ec	·#JuntosLoLogramos En el #DíaDelEstudiante, ...	49	438	115	2	33
4	Guillermo Lasso	@LassoGuillermo	·¡Felicidades @MorejonGlenda! Ella y más depor...	98	3800	620	4	59
...
303	Guillermo Lasso	@LassoGuillermo	·Tendremos Centros de Reparación Integral para...	13	283	68	1	32
304	Guillermo Lasso	@LassoGuillermo	·Fortaleceremos los espacios para las mujeres ...	10	276	61	1	22
305	Guillermo Lasso	@LassoGuillermo	·Promoveremos la conciliación de la vida famil...	41	438	124	1	38
306	Guillermo Lasso	@LassoGuillermo	·Implementaremos mecanismos de protección, den...	38	441	127	1	39
307	Guillermo Lasso	@LassoGuillermo	·Encontrémonos para luchar contra la violencia...	48	515	161	1	37

308 rows × 8 columns

Creacion del Modelo Regresion

- Obtenemos las variables de interes, en este caso el modelo sera multivariante, para valores de 'X' (comentarios, num_palabras y reacciones) y para 'Y' la salida, que seria las veces compartidas un tweet

```
In [5]: from sklearn.linear_model import LinearRegression

x=df[['comentarios','num_palabras','reacciones','hastag']]
y=df['compartido']
```

- Realizamos la division de los datos para train y test del modelo
- Utilizamos una regresion Lineal

```
In [6]: from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
import statsmodels.api as sm
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state=42)

modelR = sm.OLS(y_train, X_train).fit()
predict= modelR.predict(X_test)
```

```
In [7]: modelR.summary()
```

Time:

17:31:56

Log-Likelihood:

1769.0

No. Observations:

246

AIC:

3546.

Df Residuals:

242

BIC:

3560.

Df Model:

4

Covariance Type:

nonrobust

coef

std err

t

P>|t|

[0.025

0.975]

comentarios

1.0145

0.130

7.798

0.000

0.758

1.271

num_palabras

-1.1537

0.739

-1.561

0.120

-2.610

0.302

reacciones

0.1260

0.009

13.436

0.000

0.107

0.144

hastag

37.0617

17.531

2.114

0.036

2.529

71.595

Omnibus:

64.882

Durbin-Watson:

1.979

Prob(Omnibus):

0.000

Jarque-Bera (JB):

1557.495

Skew:

0.258

Prob(JB):

0.00

Kurtosis:

15.316

Cond. No.

5.12e+03

Notes:

[1] R² is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[3] The condition number is large, 5.12e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Creacion dataset datos predecidos

• Dataframe con los datos de test y la prediccion arrojada por nuestro modelo

• Para mas adelante poder

Notes:

[1] R² is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[3] The condition number is large, 5.12e+03. This might indicate that there are

strong multicollinearity or other numerical problems.

Creacion dataset datos predecidos

- Dataframe con los datos de test y la prediccion arrojada por nuestro modelo
- Para mas adelante poder

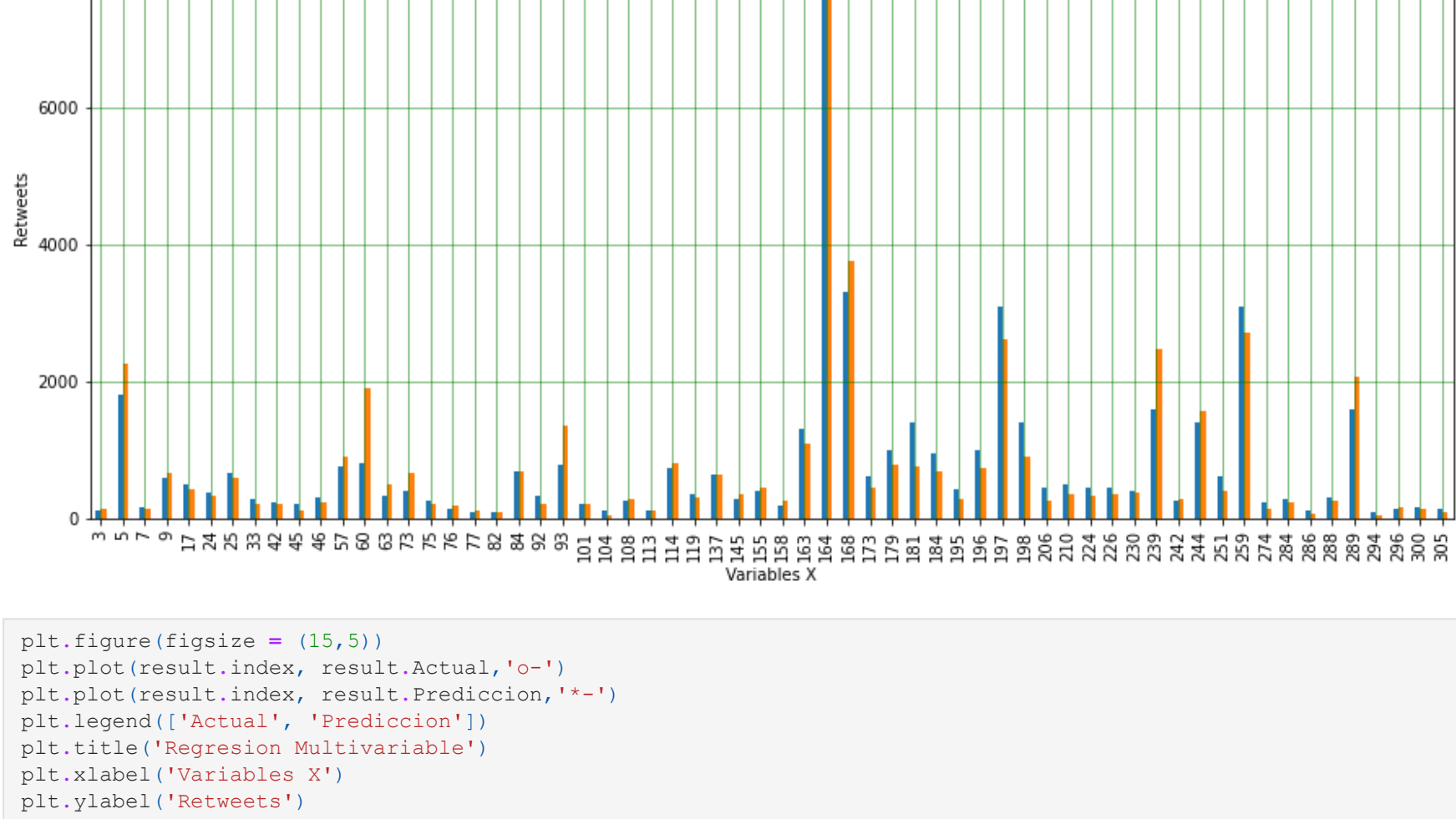
```
In [9]: result=pd.DataFrame({'Actual':y_test,'Prediccion':predict})
result.sort_index(inplace=True)
result
```

```
Out [9]:
```

	Actual	Prediccion
3	115	140.931015
5	1800	2253.892327
7	152	146.556986
9	589	667.400340
17	490	422.017938
...
289	1600	2064.038549
294	95	37.681046
296	147	154.113907
300	152	139.851137
305	124	89.984513

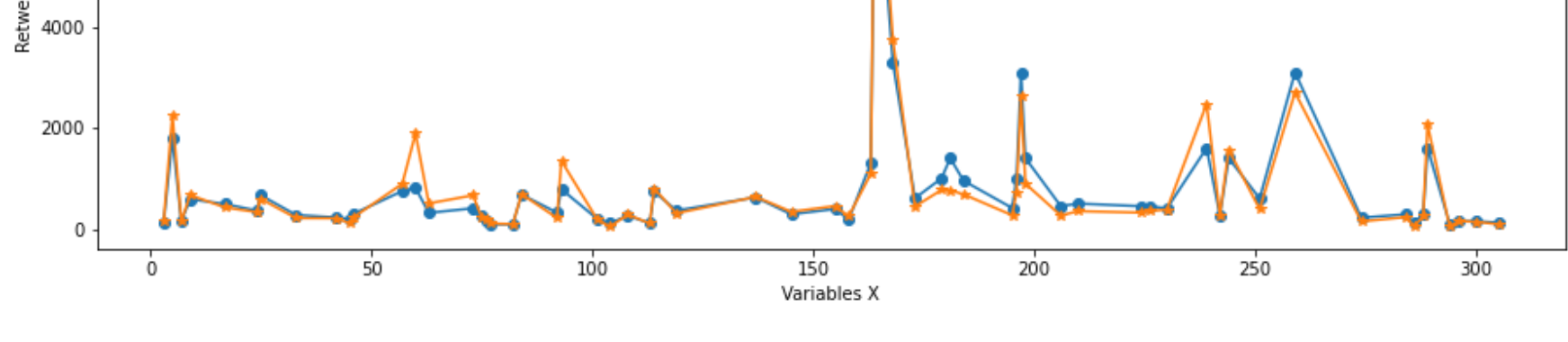
62 rows × 2 columns

```
In [10]: import matplotlib.pyplot as plt
result.plot(kind='bar',figsize=(15,7))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.title('Regresion Multivariable resultados')
plt.ylabel('Retweets')
plt.xlabel('Variables X')
plt.show()
```



```
In [11]: plt.figure(figsize = (15,5))
plt.plot(result.index, result.Actual,'o-')
plt.plot(result.index, result.Prediccion,'*-')
plt.legend(['Actual', 'Prediccion'])
plt.title('Regresion Multivariable')
plt.xlabel('Variables X')
plt.ylabel('Retweets')
plt.show()
```

```
Out [11]: Text(0, 0.5, 'Retweets')
```



Creacion Modelo Polinomial

```
In [95]: from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

poly_reg = PolynomialFeatures(degree=2)
X_poly = poly_reg.fit_transform(x)
X_train2, X_test2, y_train2, y_test2 = train_test_split(X_poly, y, test_size = 0.2, random_state=42)

pol_reg = LinearRegression()
pol_reg.fit(X_train2, y_train2)
pol_pred=pol_reg.predict(X_test2)
```

```
In [96]: result2=pd.DataFrame({'Actual':y_test2,'Prediccion':pol_pred})
result2.sort_index(inplace=True)
result2
```

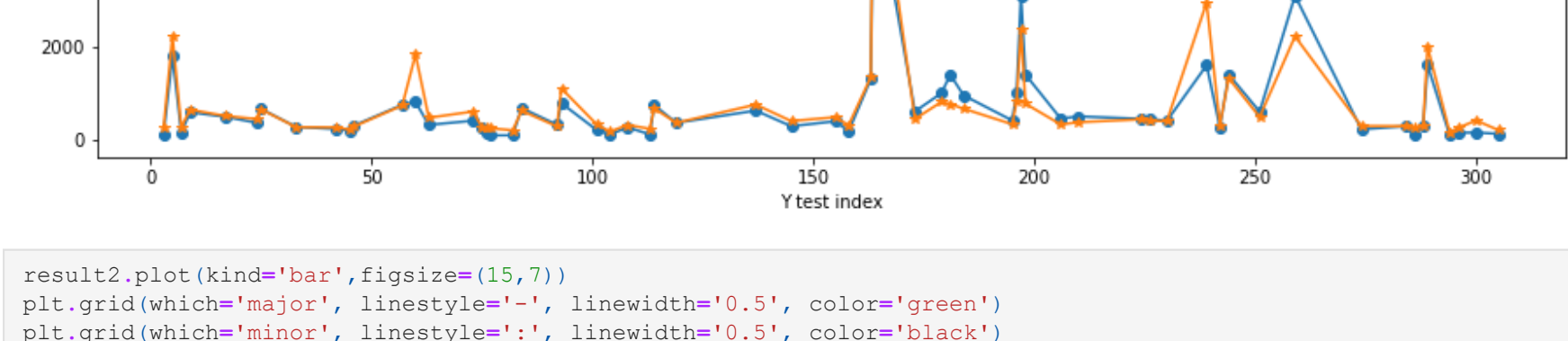
```
Out [96]:
```

	Actual	Prediccion
3	115	266.977014
5	1800	2228.460306
7	152	271.469641
9	589	645.536867
17	490	509.715460
...
289	1600	1996.207399
294	95	156.335961
296	147	257.390305
300	152	411.521624
305	124	205.959286

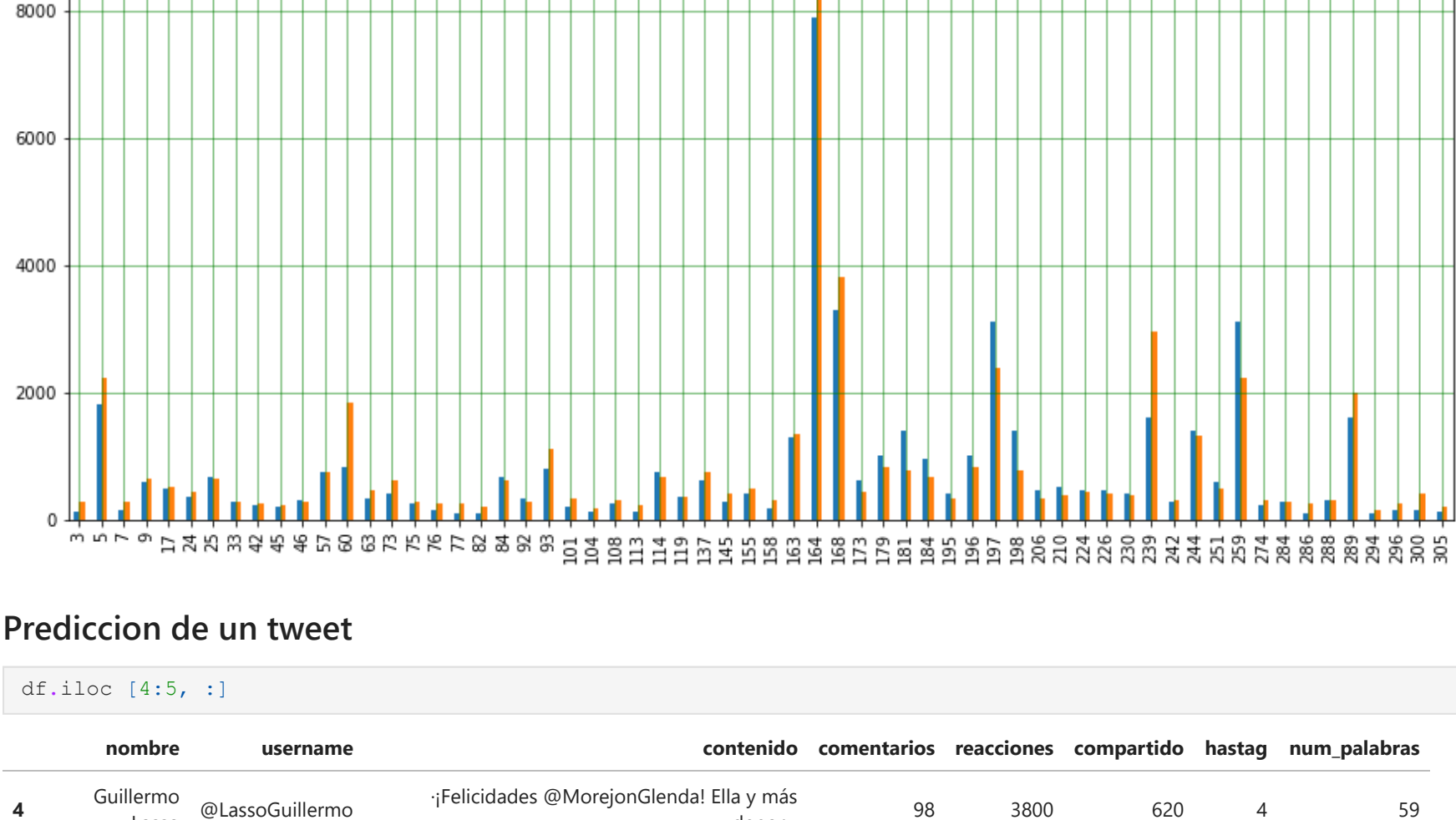
62 rows × 2 columns

```
In [97]: plt.figure(figsize = (15,5))
plt.plot(result2.index, result2.Actual,'o-')
plt.plot(result2.index, result2.Prediccion,'*-')
plt.legend(['Actual', 'Prediccion'])
plt.title('Regresion Multivariable')
plt.xlabel('Y test index')
plt.ylabel('Y test Value')
plt.show()
```

```
Out [97]: Text(0, 0.5, 'Y test Value')
```



```
In [98]: result2.plot(kind='bar',figsize=(15,7))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.title('Regresion Multivariable resultados')
plt.show()
```



Prediccion de un tweet

```
In [99]: df.iloc [4:5, :]
```

```
Out [99]:
```

	nombre	username	contenido	comentarios	reacciones	compartido	hastag	num_palabras
4	Guillermo Lasso	@LassoGuillermo	·¡Felicidades @MorejonGlenda! Ella y más depor...	98	3800	620	4	59

Regresion Lineal

```
In [106...]: print('Precision del modelo: ',r2_score(y_test,predict))
print('***( Sera compartido: '+str(int(pol_reg.predict((df.loc[4]['comentarios'],df.loc[4]['num_palabras'],
df.loc[4]['reacciones'], df.loc[4]['hastag']]]))))+' ve
```

Precision del modelo: 0.9377209236997971
(Sera compartido: 658 veces)

Regresion Polinomial

```
In [107...]: print('Precision del modelo: ',r2_score(y_test2,pol_pred))
print('***( Sera compartido: '+str(int(pol_reg.predict((poly_reg.fit_transform((df.loc[4]['comentarios'],df.loc[4]['num_palabras'],
df.loc[4]['reacciones'], df.loc[4]['hastag']]]))))+' ve
```

Precision del modelo: 0.8962906326525246
(Sera compartido: 927 veces)

Analisis

- Como se puede observar, en este caso en el que la regresion es multivariable los resultados claramente es mejor para la regresion Lineal con una mejor precision en comparacion a la regresion Polinomial, se probó incluso subiendole el grado del polinomio pero se obtuvieron peores resultados.

```
In [ ] :
```



```
In [15]: x = [1,2,3,4,5]

tags = ['Total', 'Dosis Completa', 'No Vacunados Segunda', 'Primera Dosis', 'No proceso']

fig, ax = plt.subplots(figsize=(10,5))
bar1 = ax.bar(x[0], len(estado_personas), 0.25, label='Total', color='#0070C2')
bar2 = ax.bar(x[1], vacunados_segunda, 0.25, label='Dosis Completa', color='#9318E2')
bar3 = ax.bar(x[2], no_atendidos, 0.25, label='No Vacunados', color='#1A5578')
bar4 = ax.bar(x[3], personas_nuevas, 0.25, label='Primera Dosis', color='#C21A55')
bar5 = ax.bar(x[4], no_proceso, 0.25, label='No Vacunados', color='#D70000')

ax.set_ylabel('Cantidad de personas')
ax.set_title('Personas')
ax.set_ticks(x)
ax.set_ticklabels(tags)
ax.legend(loc='upper right')
```

