

Nombre: Bárbara Rossana Pérez Silva
Grupo: 031 2015576

1 Introducción

La regresión lineal múltiple viene siendo un plus de la regresión lineal simple que vimos en la práctica anterior; aquí se vuelve a hacer uso de las variables dependientes e independientes, con la característica de que aquí se podrá hacer uso de múltiples variables independientes (x_1 , x_2 ...). Los modelos de regresión múltiple pueden emplearse para predecir el valor de la variable dependiente o para evaluar la influencia que tienen los predictores sobre ella.

2 Metodología

Para esta nueva práctica lo que haremos es mejorar el modelo que realizamos de Regresión Lineal, ahora aumentando los datos de entrada a 2 variables predictivas, para así visualizar nuestra recta, ahora plano, en un gráfico en 3D.

Para esto, necesitaremos crear una nueva variable la cual será la suma de 3 columnas de entrada: la cantidad de enlaces, comentarios y cantidad de imágenes, mientras que nuestra primer variable será igualmente la cantidad de palabras del artículo. Teniendo así 2 variables de entrada en `XY_train` y nuestra variable de salida pasa de ser “Y” a ser el eje “Z”.

```
39 suma = (filtered_data["# of Links"] + filtered_data['# of comments'].fillna(0) + filtered_data['# of images'].fillna(0))
40 dataX2 = pd.DataFrame()
41 dataX2["Word count"] = filtered_data["Word count"]
42 dataX2["suma"] = suma
43 XY_train = np.array(dataX2)
44 z_train = filtered_data['# Shares'].values
```

Para implementar la regresión lineal múltiple, creamos un nuevo objeto el cual entrenaremos con 2 dimensiones, dándonos así la posibilidad de ahora graficar el plano que buscamos. También, haremos la predicción con la que tendremos puntos sobre el plano hallado.

```
46 regr2 = linear_model.LinearRegression()
47 regr2.fit(XY_train, z_train)
48 z_pred = regr2.predict(XY_train)
```

Y por último, imprimimos los coeficientes y puntajes obtenidos.

```
50 print('Coefficients: ', regr2.coef_)
51 print("Mean squared error: %.2f" % mean_squared_error(z_train, z_pred))
52 print('Variance score: %.2f' % r2_score(z_train, z_pred))
```

Y ahora, para poder crear nuestro plano en 3D, necesitamos crear un lienzo en donde se mostrará el plano. Seguido a esto, creamos una malla de puntos en los ejes X e Y, y calculamos los valores del plano para los puntos X e Y.

```
54 fig = plt.figure()
55 ax = fig.add_subplot(111, projection='3d')
56 xx, yy = np.meshgrid(np.linspace(0, 3500, num=10), np.linspace(0, 60, num=10))
57 nuevoX = (regr2.coef_[0] * xx)
58 nuevoY = (regr2.coef_[1] * yy)
```

Calculamos los correspondientes valores para z, para ello debemos sumar el punto de intercepción, y seguido a esto colorearemos los puntos obtenidos por el entrenamiento del modelo de azul y de rojo los puntos obtenidos por la predicción de regresión lineal múltiple. Y por último, situamos el *pov* de la cámara, en donde visualizaremos el plano.

```
60 z = (nuevoX + nuevoY + regr2.intercept_)
61 ax.plot_surface(xx, yy, z, alpha=0.2, cmap='hot')
62 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue', s=30)
63 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red', s=40)
64 ax.view_init(elev=30., azimuth=65)
```

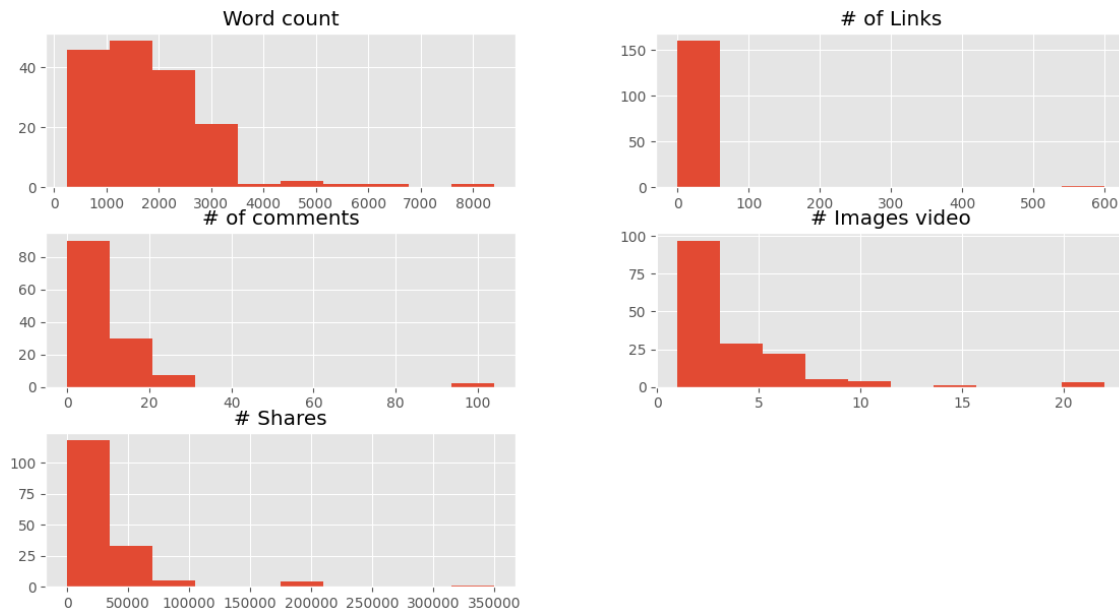
Establecemos las etiquetas que tendrá nuestra gráfica para poder guiarnos correctamente al visualizar nuestro plano, y finalmente, mostramos el plano.

```
66 ax.set_xlabel('Cantidad de Palabras')
67 ax.set_ylabel('Cantidad de Enlaces, Comentarios e Imágenes')
68 ax.set_zlabel('Compartido en Redes')
69 ax.set_title('Regresión Lineal con Múltiples Variables')
70 plt.show()
```

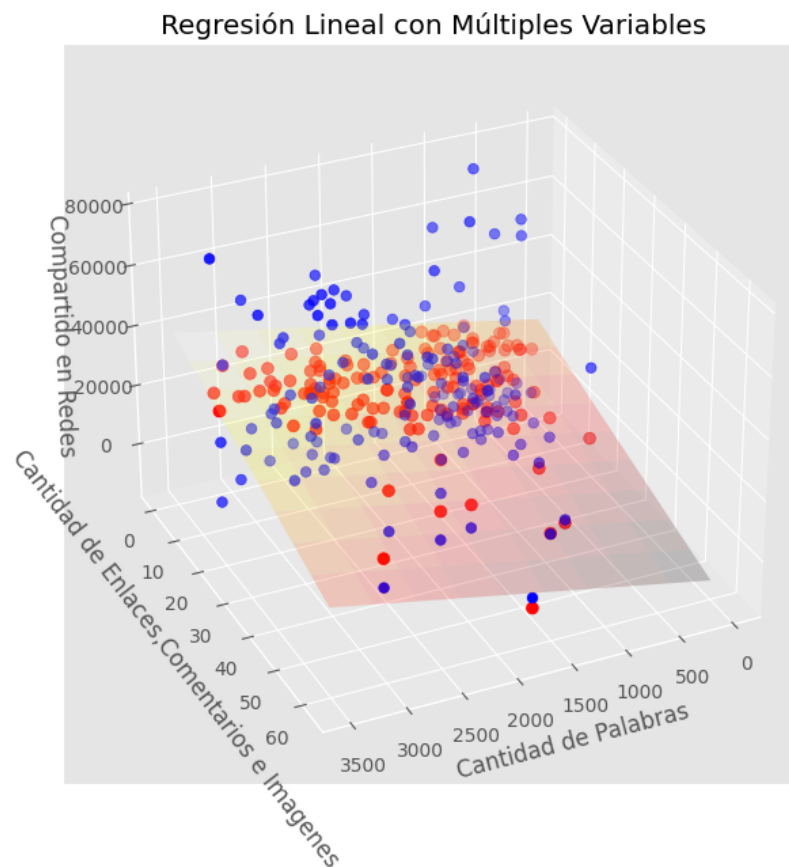
3 Resultados

```
Coefficients: [ 6.63216324 -483.40753769]
Mean squared error: 352122816.48
Variance score: 0.11
```

Ahora obtenemos dos coeficientes, de nuestras dos variables de entrada, haciendo posible utilizar la Regresión Lineal Múltiple, el error cuadrático es aún muy grande, al igual que la varianza, pero ambos resultados son mejores que los obtenidos en la práctica pasada en donde solo utilizamos una variable; igualmente, se tendría que trabajar en otro modelo para que este sea eficiente.



Podemos volver a ver las gráficas de como los datos están ordenados.



Y este sería finalmente nuestro plano en 3D creado con ayuda de la Regresión Lineal Múltiple.

Como *plus*, ejecutamos la función de predicción, nuevamente, para un artículo de 2000 palabras, con 10 enlaces, 4 comentarios y 6 imágenes. Obteniendo así, una predicción un tanto más real que la que obtuvimos en la anterior práctica con solo una variable.

```
72 z_Dosmil = regr2.predict([[2000, 10+4+6]])  
73 print('Predicción:',int(z_Dosmil[0]))
```

Predicción: 20518

4 Conclusión

Para esta práctica utilizamos el código de la anterior por lo que fue más sencillo de manipularlo, ya que ya teníamos la base para iniciar. La regresión lineal múltiple viene siendo un *upgrade* de la anterior, así que no fue tan difícil comprenderlo esta vez, y me pareció una mejor forma, ya sea de representar los datos, con el plano en 3D, y de formar predicciones.