

Basile Álvarez Andrés José

No. Cuenta: 316617187

Email: andresbasile123@gmail.com

Fecha: 9/10/21

Inteligencia Artificial

Grupo III

Semestre 2022-1

## Reporte Práctica 7: Pronóstico con regresión lineal múltiple

**Objetivo:** Obtener el pronóstico de la saturación de aceite remanente (ROS, Residual Oil Saturation) a partir de las cuatro mediciones de los registros geofísicos convencionales: (RC1) Registro Neutrón, (RC2) Registro Sónico, (RC3) Registro Densidad-Neutrón, y (RC4) Registro Densidad Corregido por Arcilla.

**Fuente de datos:** Mediciones de registros geofísicos convencionales: : (RC1) Registro Neutrón, (RC2) Registro Sónico, (RC3) Registro Densidad-Neutrón, y (RC4) Registro Densidad Corregido por Arcilla.

### Características Generales:

En esta práctica, utilizaremos la regresión lineal múltiple (trabajando con cinco variables: las cuatro mencionadas anteriormente [de la cual la que más nos interesa es el RC4] y la profundidad en pies). A lo largo del algoritmo, utilizamos a la variable RC4 como variable a pronostica y a las otras variables como variables predictoras.

La regresión lineal es una forma de aprendizaje supervisado que permite encontrar una ecuación que minimiza la distancia entre la línea ajustada y los puntos de los datos. Esto nos permitirá hacer pronósticos de qué valores tomará una variable para distintos puntos.

### Desarrollo

Primeramente, tenemos que definir aquellas bibliotecas de Python que nos serán útiles para importar, limpiar y analizar los datos contenidos en el archivo separado por comas *RGeofisicos.csv*. Estas serán: *pandas* (manipulación y análisis de datos), *matplotlib* (para la creación de gráficas y visualización de los datos), *numpy* para utilizar vectores y matrices de  $n$  dimensiones.

Más tarde, importamos el archivo *RGeofisicos.csv* y lo primero que hacemos es guardarlo en un DataFrame. La importación del archivo se realizó a partir del explorador de archivos que abrimos utilizando el comando *files.upload()*. Una vez importados los datos, mostramos el DataFrame que los contiene:

```
RGeofisicos = pd.read_csv('RGeofisicos.csv')
RGeofisicos
```

	Profundidad	RC1	RC2	RC3	RC4
0	5660.0	0.777924	0.814029	0.675698	0.757842
1	5660.5	0.796239	0.813167	0.748670	0.793872
2	5661.0	0.769231	0.797562	0.702285	0.748362
3	5661.5	0.764774	0.790365	0.680289	0.738451
4	5662.0	0.773813	0.788184	0.700248	0.718462
5	5662.5	0.795627	0.798850	0.753472	0.777537
6	5663.0	0.802155	0.837717	0.785441	0.807957
7	5663.5	0.797878	0.833851	0.756847	0.779641
8	5664.0	0.777206	0.813117	0.718713	0.761454
9	5664.5	0.788604	0.820041	0.729582	0.765600
10	5665.0	0.776924	0.815917	0.737350	0.788688
11	5665.5	0.769003	0.797940	0.724736	0.779675
12	5666.0	0.755305	0.815150	0.679189	0.762972
13	5666.5	0.746095	0.804713	0.659602	0.754690
14	5667.0	0.757050	0.793180	0.651374	0.748380
15	5667.5	0.744187	0.786476	0.612430	0.688062
16	5668.5	0.747083	0.798745	0.674513	0.714754
17	5669.0	0.752375	0.785494	0.711418	0.753766
18	5669.5	0.733356	0.779964	0.683226	0.727931
19	5670.0	0.713796	0.769322	0.600747	0.682140

Figura 1: Data Frame con algunos de los datos geofísicos.

Una vez hecho lo anterior, hacemos una gráfica de los registros geofísicos convencionales a analizar utilizando *matplotlib*. En la gráfica presentamos los registros RC1, RC2, RC3 y RC4 definidos anteriormente en porcentaje, colocando en el eje horizontal la profundidad en pies de las mediciones para dichos registros.

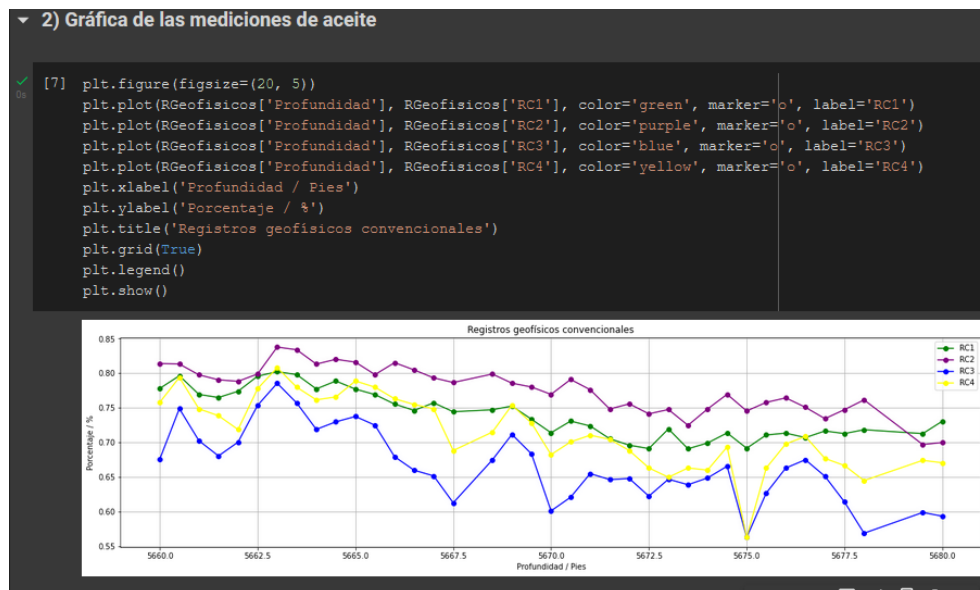


Figura 2: Gráfica de los registros geofísicos en función de la profundidad en pies.

Para la aplicación del algoritmo de regresión lineal, primero tenemos que hacer la importación de las funciones de *linear\_model* y *mean\_squared\_error*, *max\_error*, *r2\_score* de la biblioteca *sklearn* y *sklearn.metrics*, respectivamente. La función *mean\_squared\_error* nos permitirá calcular qué tanto difieren los valores estimados de los valores reales, el *max\_error* nos permitirá encontrar el residuo de la función, mientras que el *r2\_score* será la bondad del ajuste a utilizar.

Primeramente, tenemos que seleccionar las variables predictoras y la variable a pronosticar. Con tal fin, primero hacemos una matriz con *numpy* en donde colocamos las variables *Profundidad*, *RC1*, *RC2* y *RC3*, las cuales nos servirán como variables predictoras.

```
[10] X_train = np.array(RGeofisicos[['Profundidad', 'RC1', 'RC2', 'RC3']])
      pd.DataFrame(X_train)
```

	0	1	2	3
0	5660.0	0.777924	0.814029	0.675698
1	5660.5	0.796239	0.813167	0.748670
2	5661.0	0.769231	0.797562	0.702285
3	5661.5	0.764774	0.790365	0.680289
4	5662.0	0.773813	0.788184	0.700248
5	5662.5	0.795627	0.798850	0.753472
6	5663.0	0.802155	0.837717	0.785441
7	5663.5	0.797878	0.833851	0.756847
8	5664.0	0.777206	0.813117	0.718713
9	5664.5	0.788604	0.820041	0.729582
10	5665.0	0.776924	0.815917	0.737350
11	5665.5	0.769003	0.797940	0.724736
12	5666.0	0.755305	0.815150	0.679189
13	5666.5	0.746095	0.804713	0.659602
14	5667.0	0.757050	0.793180	0.651374

Figura 3: Parte de la matriz con las variables predictoras.

Para la variable a predecir, utilizaremos *RC4*, la cual colocamos en una matriz diferente, como se muestra en la figura 4.

Figura 4: Parte de la matriz con la variable a predecir.

```
Y_train = np.array(RGeofisicos[['RC4']])
pd.DataFrame(Y_train)
```

	0
0	0.757842
1	0.793872
2	0.748362
3	0.738451
4	0.718462
5	0.777537
6	0.807957

En el punto anterior, buscamos aproximar a un valor único y, luego de realizar lo anterior, hacemos el entrenamiento del modelo utilizando una regresión lineal múltiple. Para ello, llamamos a la función *LinearRegression* de *linear\_model*, pasando como argumentos de la función *fit* a la variable a predecir y las variables predictoras.

Hecho lo anterior, generamos el pronóstico para los valores de *X\_train* (variables predictoras) y colocamos el resultado en un *DataFrame* nuevo, como se muestra en la figura 5.

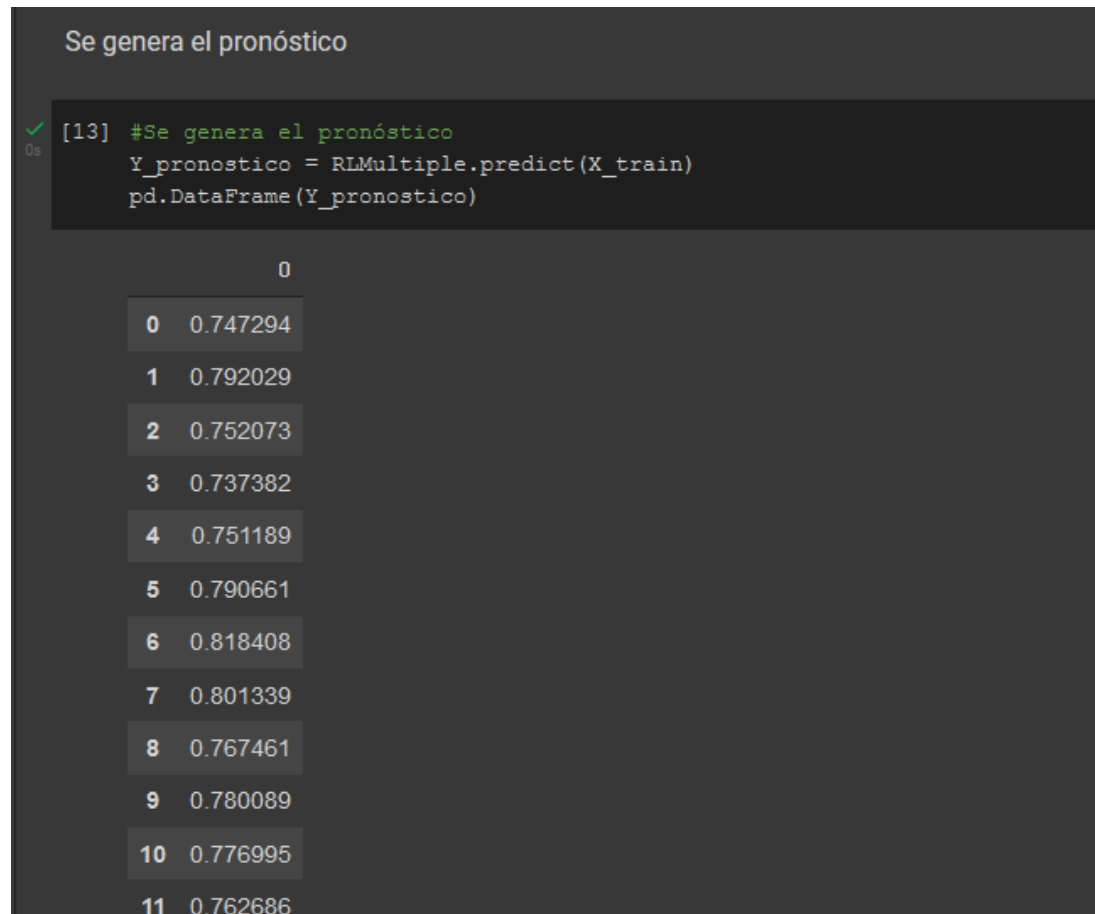
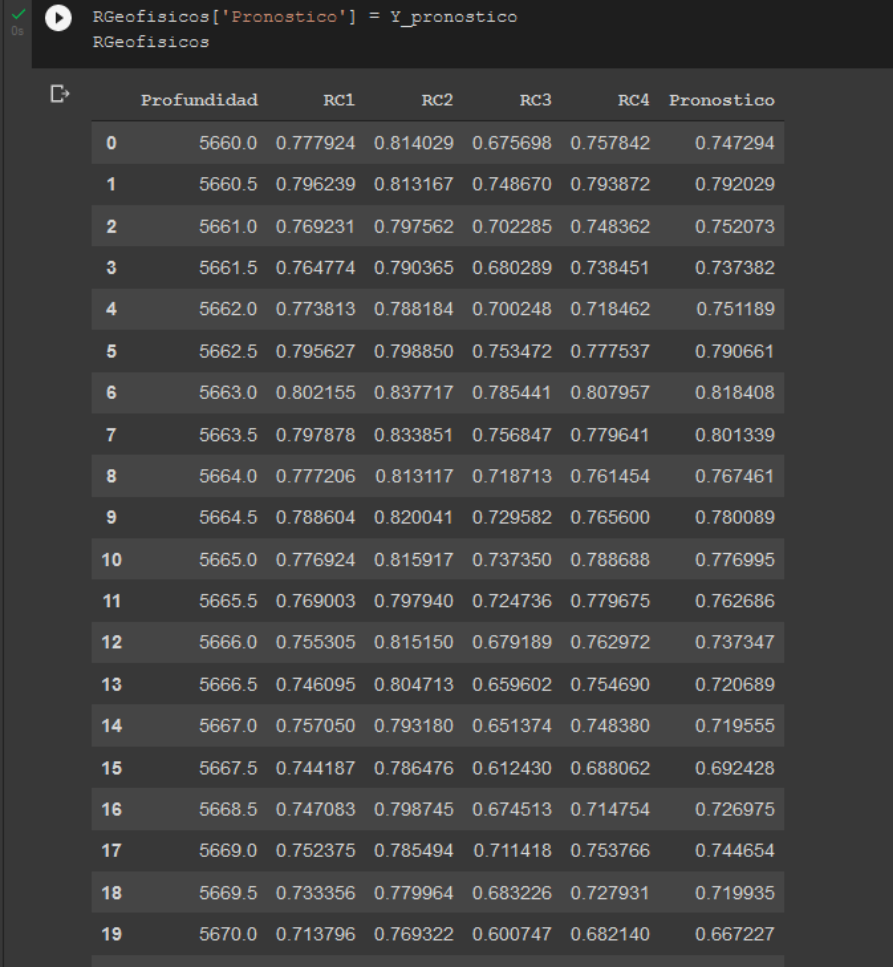


Figura 5: Parte del pronóstico generado.

El pronóstico generado es agregado al *DataFrame* *RGeofisicos* en la columna *Pronostico*. Al imprimir este *DataFrame*, observamos los valores de las variables predictoras y el valor del pronóstico para cada una de ellas, lo cual nos permite tener una noción general de los valores que se podrían obtener para *RC4* y compararlo con el valor real que teníamos en los datos originales. Dicho *DataFrame* se muestra en la figura 6.



The screenshot shows an RStudio interface. At the top, a code editor displays the command `RGeofisicos['Pronostico'] = Y_pronostico` followed by `RGeofisicos`. Below the code, a data frame table is displayed with 20 rows and 7 columns. The columns are labeled: Profundidad, RC1, RC2, RC3, RC4, and Pronostico. The rows are indexed from 0 to 19. The 'Pronostico' column contains values ranging from approximately 0.667 to 0.792.

	Profundidad	RC1	RC2	RC3	RC4	Pronostico
0	5660.0	0.777924	0.814029	0.675698	0.757842	0.747294
1	5660.5	0.796239	0.813167	0.748670	0.793872	0.792029
2	5661.0	0.769231	0.797562	0.702285	0.748362	0.752073
3	5661.5	0.764774	0.790365	0.680289	0.738451	0.737382
4	5662.0	0.773813	0.788184	0.700248	0.718462	0.751189
5	5662.5	0.795627	0.798850	0.753472	0.777537	0.790661
6	5663.0	0.802155	0.837717	0.785441	0.807957	0.818408
7	5663.5	0.797878	0.833851	0.756847	0.779641	0.801339
8	5664.0	0.777206	0.813117	0.718713	0.761454	0.767461
9	5664.5	0.788604	0.820041	0.729582	0.765600	0.780089
10	5665.0	0.776924	0.815917	0.737350	0.788688	0.776995
11	5665.5	0.769003	0.797940	0.724736	0.779675	0.762686
12	5666.0	0.755305	0.815150	0.679189	0.762972	0.737347
13	5666.5	0.746095	0.804713	0.659602	0.754690	0.720689
14	5667.0	0.757050	0.793180	0.651374	0.748380	0.719555
15	5667.5	0.744187	0.786476	0.612430	0.688062	0.692428
16	5668.5	0.747083	0.798745	0.674513	0.714754	0.726975
17	5669.0	0.752375	0.785494	0.711418	0.753766	0.744654
18	5669.5	0.733356	0.779964	0.683226	0.727931	0.719935
19	5670.0	0.713796	0.769322	0.600747	0.682140	0.667227

Figura 6: Parte del *DataFrame* con el pronóstico generado.

Más tarde, utilizamos distintas métricas para conocer la efectividad de nuestro modelo. Dichos valores para calcular son el *MSE*, el *punto de corte*, el *score*, entre otros. Con el intercepto, logramos conocer el punto en donde el modelo obtenido cruzará el eje de las ordenadas, el error o residuo nos permitirá conocer la variación entre los valores del pronóstico y los valores originales (el residuo fue bajo, con un valor de 0.068, que nos indica que los valores pronosticados se acercan a los reales). Luego, calculamos el *Score* o bondad de ajuste, donde obtuvimos un valor de 0.8581 que indica que se tuvo una bondad de ajuste de 85%, la cual es considerada buena. Los valores obtenidos se presentan en la figura 7.

Figura 7: Valores de los coeficientes, intercepto, error y score para el modelo generado.

```

Coeficientes:
[[-7.50589329e-05  5.06619053e-01  2.27471256e-01  4.89091335e-01]]
Intercepto:
[0.26237022]
Residuo: 0.0684
MSE: 0.0004
RMSE: 0.0195
Score (Bondad de ajuste): 0.8581

```

Posteriormente, obtuvimos el modelo matemático para el ejemplo utilizado. Esto es equivalente a obtener la recta que minimiza la distancia entre la línea ajustada y los puntos de los datos. Para el modelo de pronóstico, obtenemos:

$$Y = 0.2624 - 0.000075(\text{Profundidad}) + 0.5066(RC1) + 0.2275(RC2) + 0.4891(RC3) + 0.0684$$

- Se tiene un Score (Bondad de ajuste) de 0.8581, el cual indica que el pronóstico de la saturación de aceite remanente (SOR), en un determinado nivel de profundidad, se logrará con un 85.81% de efectividad (grado de intensidad).
- Además, los pronósticos del modelo final se alejan en promedio 0.0004 y 0.0195 unidades del valor real, esto es, MSE y RMSE, respectivamente.

Figura 8: Conformación del modelo de pronóstico.

Una vez realizado lo anterior, obtenemos la gráfica comparativa entre los valores reales y los valores pronosticados. Utilizaremos la misma gráfica que habíamos impreso anteriormente, agregando los puntos de la variable *RC4* pronosticada, la cual observamos que estima de manera bastante acertada los valores reales.

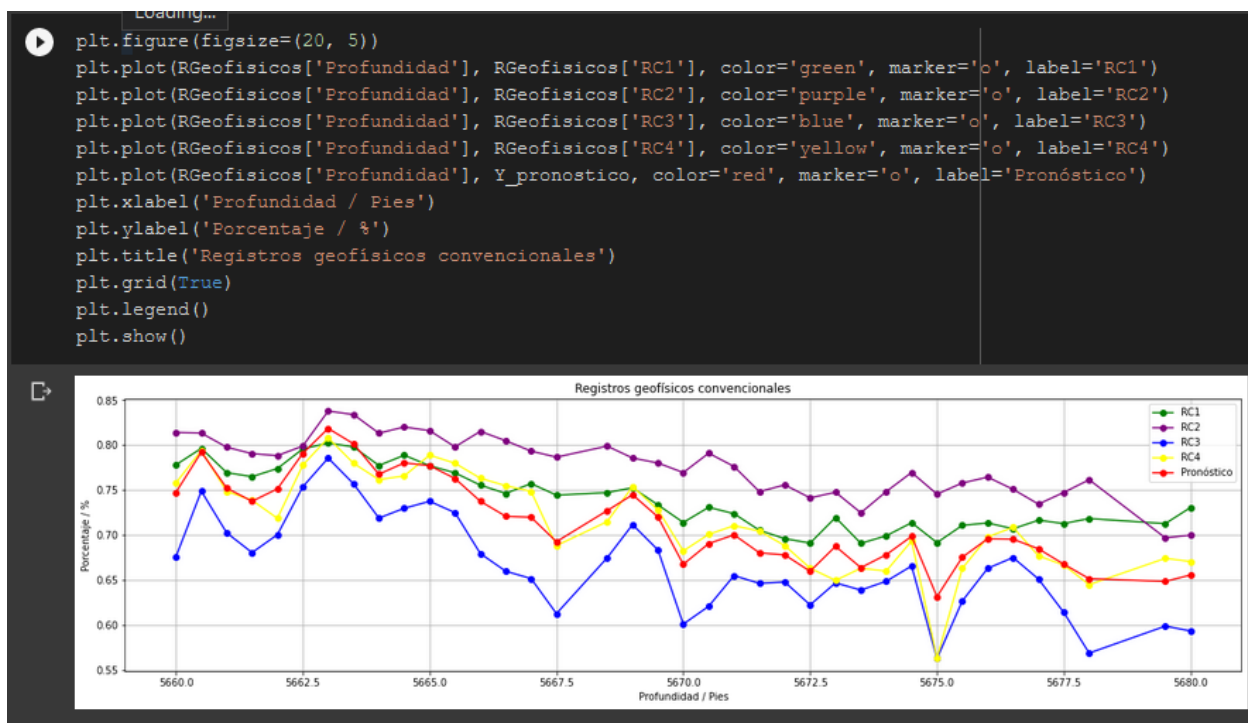


Figura 9: Comparativa entre los valores reales y los del modelo de pronóstico.

Finalmente, podemos obtener pronósticos para la variable *RC4* a partir del modelo generado, ingresando los valores de las variables *Profundidad*, *RC1*, *RC2* y *RC3*. Esto nos será útil para encontrar la correspondencia de *RC4* para valores nuevos de las variables predictoras, permitiendo que el modelo dé valores para situaciones fuera de las tomadas en los datos originales. No obstante, es conveniente que los rangos de las variables predictoras se encuentren cercanos o dentro de los rangos originales, para así evitar resultados erróneos o pronósticos muy alejados de lo que realmente sucedería.

## **Conclusiones**

A lo largo de esta práctica, tuvimos como principal objetivo el obtener el pronóstico de una variable geofísica a partir de una regresión lineal múltiple con 4 variables predictoras. Esto es un algoritmo de aprendizaje supervisado.

Al utilizar la regresión lineal múltiple, logramos hacer un pronóstico a partir de los registros geofísicos convencionales y obtener un modelo matemático que nos permita encontrar valores pronosticados para una variable geofísica a partir de la obtención de otros datos geofísicos.

Este tipo de algoritmo es muy utilizado en diversas aplicaciones. Si bien en esta práctica utilizamos este tipo de aprendizaje para datos geofísicos, esto se podría traducir a datos de todo tipo como para obtener valores pronosticados del valor de una acción a partir de otras variables económicas, entre otros ejemplos.

En general, creo que se cumplieron los objetivos de la práctica y que ésta fue útil para comprender de mejor manera un ejemplo de aprendizaje supervisado sencillo y las posibles aplicaciones de este tipo de algoritmos.