

Universidad Nacional Autónoma de México
Facultad de Ingeniería

Inteligencia Artificial

PRÁCTICA 7. PRONÓSTICO CON REGRESIÓN LINEAL MÚLTIPLE

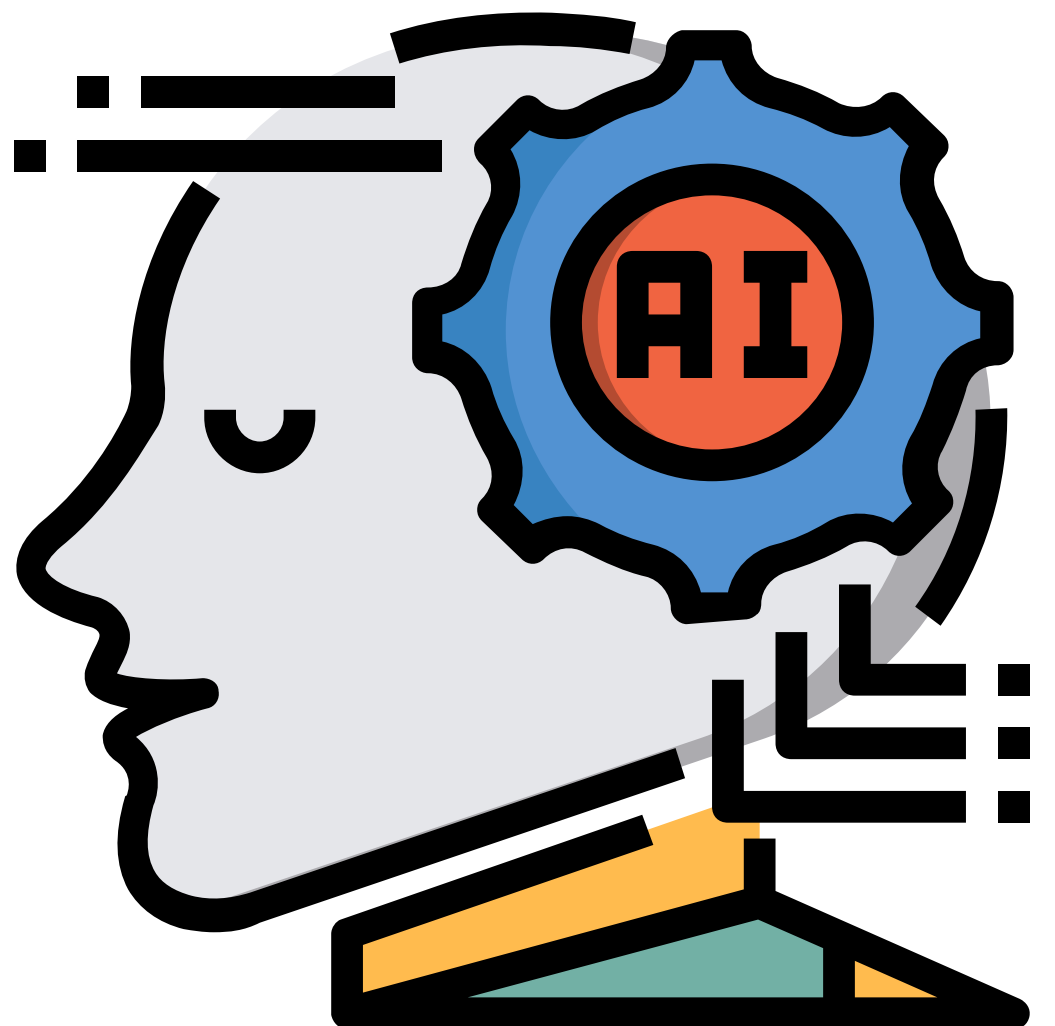


Casasola García Oscar

316123747

oscar.casasola.g7@gmail.com

Grupo 03



Profesor: Dr. Guillermo Gilberto Molero Castillo
Semestre 2022-1

Contenido

Contexto 2

 Objetivo 2

 Fuente de datos 2

Preparación del entorno de ejecución..... 2

 1) Importar las bibliotecas necesarias 2

 2) Importar los datos..... 2

Gráfica de las mediciones de aceite 3

Aplicación del algoritmo 3

 Se seleccionan las variables predictoras (X) y la variable a pronosticar (Y) 3

 Se entrena el modelo a través de una regresión lineal múltiple 4

 Se genera el pronóstico..... 4

 Valores pronosticados 5

Obtención de los coeficientes, intercepto, error y Score 5

Comparación del modelo de pronóstico 5

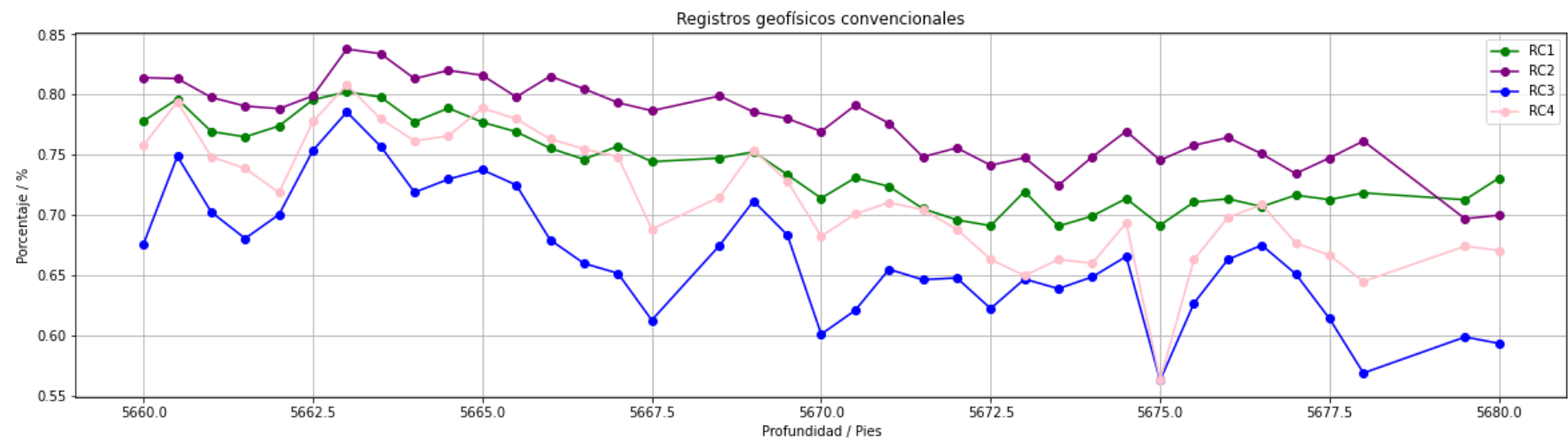
Proyección de los valores reales y pronosticados 6

Nuevos pronósticos..... 6

Conclusiones 6

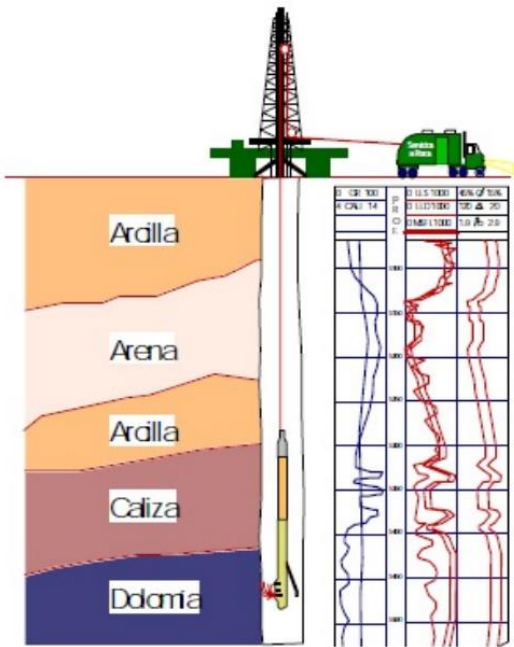
Contexto

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

Se tienen mediciones de registros geofísicos convencionales: RC1 (Registro Neutrón), RC2 (Registro Sónico), RC3 (Registro Densidad-Neutrón) y RC4 (Registro Densidad -corregido por arcilla-)



- Para la toma de registros se cuenta con cables electromecánicos, sensores, dispositivos eléctricos y sistemas computarizados.
- Se procesan los datos a través de los sensores, para luego ser enviados a la superficie por medio del cable.

- ⇒ RC1 = Registro Neutrón
- ⇒ RC2 = Registro Sónico
- ⇒ RC3 = Registro Densidad-Neutrón
- ⇒ RC4 = Registro Densidad (corregido por arcilla)

Preparación del entorno de ejecución

1) Importar las bibliotecas necesarias

```
import pandas as pd # Para la manipulación y análisis de datos
import numpy as np # Para crear vectores y matrices n dimensionales
import matplotlib.pyplot as plt # Para la generación de gráficas a partir de los datos
import seaborn as sns # Para la visualización de datos basado en matplotlib
%matplotlib inline
```

2) Importar los datos

Fuente de datos: RGeofisicos.csv

```
# Si se usa Google Colab
#from google.colab import files
#files.upload()

#from google.colab import drive
#drive.mount('/content/drive')
```

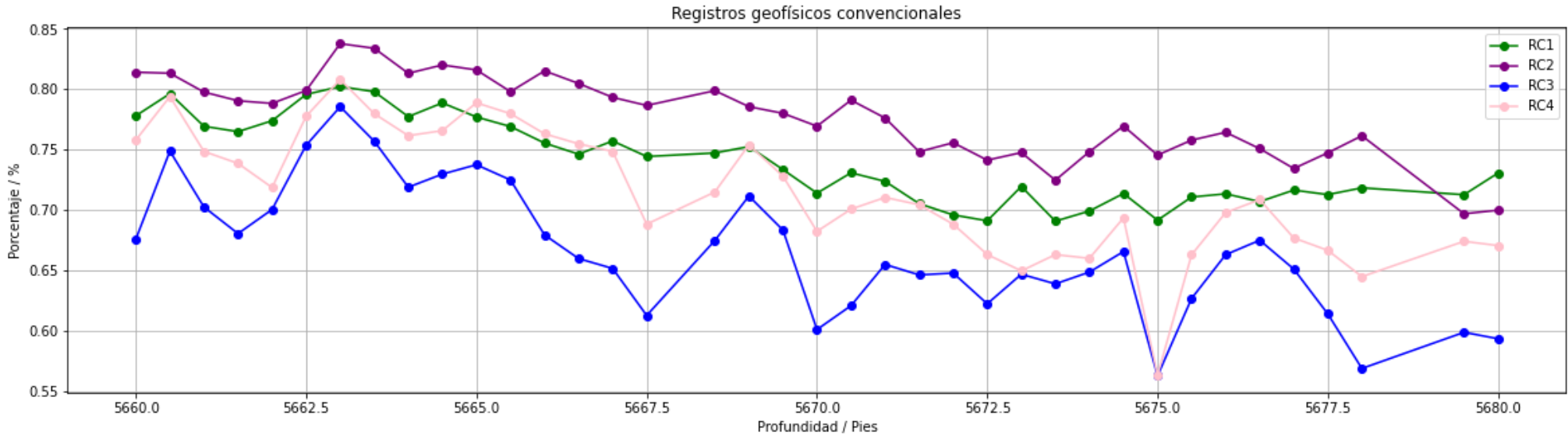
```
RGeofisicos = pd.read_csv("RGeofisicos.csv")
RGeofisicos
```

	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
20	5670.5	0.730675	0.790874	0.620547	0.700536
21	5671.0	0.723575	0.776079	0.654705	0.710247
22	5671.5	0.705104	0.748151	0.646052	0.704211
23	5672.0	0.695764	0.755593	0.647573	0.688053
24	5672.5	0.690900	0.741153	0.622093	0.662947
25	5673.0	0.719354	0.747426	0.646606	0.649385
26	5673.5	0.690589	0.724560	0.638601	0.662852
27	5674.0	0.698980	0.748176	0.648402	0.659771
28	5674.5	0.713517	0.769434	0.665381	0.693296
29	5675.0	0.691382	0.745327	0.562401	0.562902
30	5675.5	0.710586	0.757745	0.626525	0.662724
31	5676.0	0.713261	0.764235	0.662805	0.697686
32	5676.5	0.706780	0.750652	0.674763	0.708427
33	5677.0	0.716311	0.734232	0.650890	0.676664
34	5677.5	0.712586	0.747234	0.613715	0.666385
35	5678.0	0.718184	0.761320	0.568504	0.644622
36	5679.5	0.712518	0.696722	0.598481	0.673889
37	5680.0	0.730503	0.699670	0.593132	0.670354

Gráfica de las mediciones de aceite

```
plt.figure(figsize=(20,5))
plt.plot(RGeofisicos['Profundidad'],RGeofisicos['RC1'],color='green',marker='o',label='RC1')
plt.plot(RGeofisicos['Profundidad'],RGeofisicos['RC2'],color='purple',marker='o',label='RC2')
plt.plot(RGeofisicos['Profundidad'],RGeofisicos['RC3'],color='blue',marker='o',label='RC3')
plt.plot(RGeofisicos['Profundidad'],RGeofisicos['RC4'],color='pink',marker='o',label='RC4')
plt.xlabel('Profundidad / Pies')
plt.ylabel('Porcentaje / %')
plt.title('Registros geofísicos convencionales')
plt.grid(True)
plt.legend()
plt.show()
```



Aplicación del algoritmo

Se importan las bibliotecas que se van a ocupar:

```
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, max_error, r2_score
```

Se seleccionan las variables predictoras (X) y la variable a pronosticar (Y)

```
X_train = np.array(RGeofisicos[['Profundidad','RC1','RC2','RC3']]) # Separamos las variables predictoras
pd.DataFrame(X_train)
```


	0	1	2	3	19	5670.0	0.713796	0.769322	0.600747
0	5660.0	0.777924	0.814029	0.675698	20	5670.5	0.730675	0.790874	0.620547
1	5660.5	0.796239	0.813167	0.748670	21	5671.0	0.723575	0.776079	0.654705
2	5661.0	0.769231	0.797562	0.702285	22	5671.5	0.705104	0.748151	0.646052
3	5661.5	0.764774	0.790365	0.680289	23	5672.0	0.695764	0.755593	0.647573
4	5662.0	0.773813	0.788184	0.700248	24	5672.5	0.690900	0.741153	0.622093
5	5662.5	0.795627	0.798850	0.753472	25	5673.0	0.719354	0.747426	0.646606
6	5663.0	0.802155	0.837717	0.785441	26	5673.5	0.690589	0.724560	0.638601
7	5663.5	0.797878	0.833851	0.756847	27	5674.0	0.698980	0.748176	0.648402
8	5664.0	0.777206	0.813117	0.718713	28	5674.5	0.713517	0.769434	0.665381
9	5664.5	0.788604	0.820041	0.729582	29	5675.0	0.691382	0.745327	0.562401
10	5665.0	0.776924	0.815917	0.737350	30	5675.5	0.710586	0.757745	0.626525
11	5665.5	0.769003	0.797940	0.724736	31	5676.0	0.713261	0.764235	0.662805
12	5666.0	0.755305	0.815150	0.679189	32	5676.5	0.706780	0.750652	0.674763
13	5666.5	0.746095	0.804713	0.659602	33	5677.0	0.716311	0.734232	0.650890
14	5667.0	0.757050	0.793180	0.651374	34	5677.5	0.712586	0.747234	0.613715
15	5667.5	0.744187	0.786476	0.612430	35	5678.0	0.718184	0.761320	0.568504
16	5668.5	0.747083	0.798745	0.674513	36	5679.5	0.712518	0.696722	0.598481
17	5669.0	0.752375	0.785494	0.711418	37	5680.0	0.730503	0.699670	0.593132
18	5669.5	0.733356	0.779964	0.683226					

```
Y_train = np.array(RGeofisicos['RC4']) # Separamos la variable a pronosticar
pd.DataFrame(Y_train)
```

	0	19	0.682140
0	0.757842	20	0.700536
1	0.793872	21	0.710247
2	0.748362	22	0.704211
3	0.738451	23	0.688053
4	0.718462	24	0.662947
5	0.777537	25	0.649385
6	0.807957	26	0.662852
7	0.779641	27	0.659771
8	0.761454	28	0.693296
9	0.765600	29	0.562902
10	0.788688	30	0.662724
11	0.779675	31	0.697686
12	0.762972	32	0.708427
13	0.754690	33	0.676664
14	0.748380	34	0.666385
15	0.688062	35	0.644622
16	0.714754	36	0.673889
17	0.753766	37	0.670354
18	0.727931		

Se entrena el modelo a través de una regresión lineal múltiple

```
RLMultiple = linear_model.LinearRegression()
RLMultiple.fit(X_train,Y_train) # Se entrena el modelo

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

Se genera el pronóstico

```
Y_pronostico = RLMultiple.predict(X_train)
pd.DataFrame(Y_pronostico)
```

	0	19	0.667227
0	0.747294	20	0.690328
1	0.792029	21	0.700034
2	0.752073	22	0.680054
3	0.737382	23	0.677721
4	0.751189	24	0.659473
5	0.790661	25	0.687266
6	0.818408	26	0.663540
7	0.801339	27	0.677919
8	0.767461	28	0.698386
9	0.780089	29	0.631284
10	0.776995	30	0.675163
11	0.762686	31	0.695701
12	0.737347	32	0.695139
13	0.720689	33	0.684519
14	0.719555	34	0.667370
15	0.692428	35	0.651260
16	0.726975	36	0.648244
17	0.744654	37	0.655373
18	0.719935		

Valores pronosticados

RGeofisicos['Pronostico'] = Y_pronostico															
RGeofisicos															
	Profundidad	RC1	RC2	RC3	RC4	Pronostico		19	5670.0	0.713796	0.769322	0.600747	0.682140	0.667227	
0	5660.0	0.777924	0.814029	0.675698	0.757842	0.747294	20	5670.5	0.730675	0.790874	0.620547	0.700536	0.690328		
1	5660.5	0.796239	0.813167	0.748670	0.793872	0.792029	21	5671.0	0.723575	0.776079	0.654705	0.710247	0.700034		
2	5661.0	0.769231	0.797562	0.702285	0.748362	0.752073	22	5671.5	0.705104	0.748151	0.646052	0.704211	0.680054		
3	5661.5	0.764774	0.790365	0.680289	0.738451	0.737382	23	5672.0	0.695764	0.755593	0.647573	0.688053	0.677721		
4	5662.0	0.773813	0.788184	0.700248	0.718462	0.751189	24	5672.5	0.690900	0.741153	0.622093	0.662947	0.659473		
5	5662.5	0.795627	0.798850	0.753472	0.777537	0.790661	25	5673.0	0.719354	0.747426	0.646606	0.649385	0.687266		
6	5663.0	0.802155	0.837717	0.785441	0.807957	0.818408	26	5673.5	0.690589	0.724560	0.638601	0.662852	0.663540		
7	5663.5	0.797878	0.833851	0.756847	0.779641	0.801339	27	5674.0	0.698980	0.748176	0.648402	0.659771	0.677919		
8	5664.0	0.777206	0.813117	0.718713	0.761454	0.767461	28	5674.5	0.713517	0.769434	0.665381	0.693296	0.698386		
9	5664.5	0.788604	0.820041	0.729582	0.765600	0.780089	29	5675.0	0.691382	0.745327	0.562401	0.562902	0.631284		
10	5665.0	0.776924	0.815917	0.737350	0.788688	0.776995	30	5675.5	0.710586	0.757745	0.626525	0.662724	0.675163		
11	5665.5	0.769003	0.797940	0.724736	0.779675	0.762686	31	5676.0	0.713261	0.764235	0.662805	0.697686	0.695701		
12	5666.0	0.755305	0.815150	0.679189	0.762972	0.737347	32	5676.5	0.706780	0.750652	0.674763	0.708427	0.695139		
13	5666.5	0.746095	0.804713	0.659602	0.754690	0.720689	33	5677.0	0.716311	0.734232	0.650890	0.676664	0.684519		
14	5667.0	0.757050	0.793180	0.651374	0.748380	0.719555	34	5677.5	0.712586	0.747234	0.613715	0.666385	0.667370		
15	5667.5	0.744187	0.786476	0.612430	0.688062	0.692428	35	5678.0	0.718184	0.761320	0.568504	0.644622	0.651260		
16	5668.5	0.747083	0.798745	0.674513	0.714754	0.726975	36	5679.5	0.712518	0.696722	0.598481	0.673889	0.648244		
17	5669.0	0.752375	0.785494	0.711418	0.753766	0.744654	37	5680.0	0.730503	0.699670	0.593132	0.670354	0.655373		
18	5669.5	0.733356	0.779964	0.683226	0.727931	0.719935									

Obtención de los coeficientes, intercepto, error y Score

```
print('Coeficientes: \n',RLMultiple.coef_)
print('Intercepto: \n',RLMultiple.intercept_)
print('Residuo: %.4f' % max_error(Y_train,Y_pronostico))
print('MSE: %.4f' % mean_squared_error(Y_train,Y_pronostico))
print('RMSE: %.4f' % mean_squared_error(Y_train,Y_pronostico,squared=False)) #True devuelve MSE y False devuelve RMSE
print('Score (Bondad de ajuste): %.4f' % r2_score(Y_train,Y_pronostico))
```

Coeficientes:

[-7.50589329e-05 5.06619053e-01 2.27471256e-01 4.89091335e-01]

Intercepto:

0.26237021738651867

Residuo: 0.0684

MSE: 0.0004

RMSE: 0.0195

Score (Bondad de ajuste): 0.8581

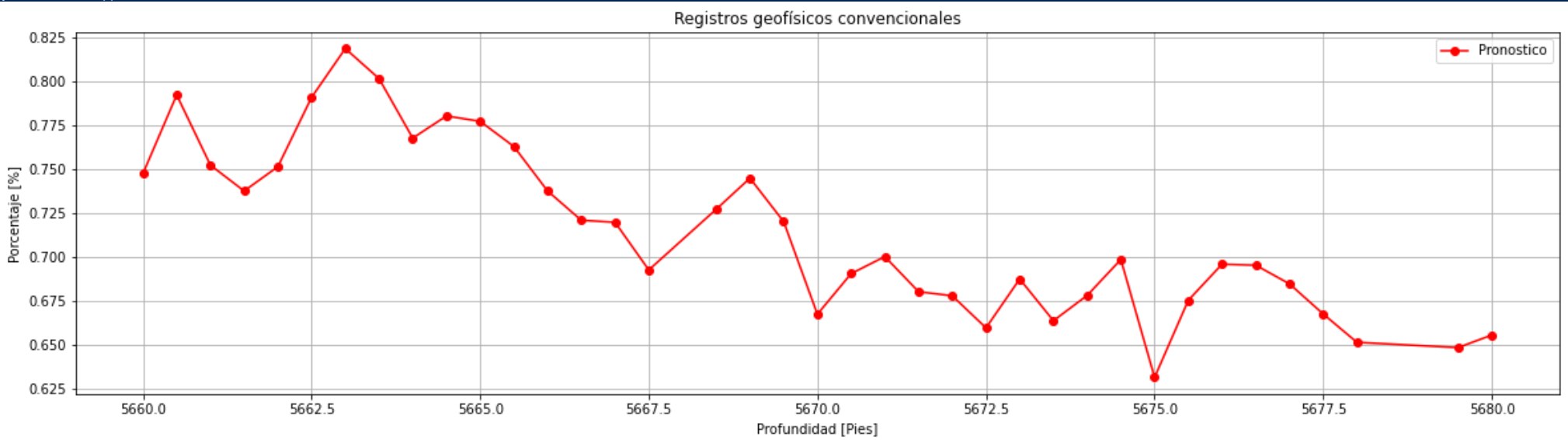
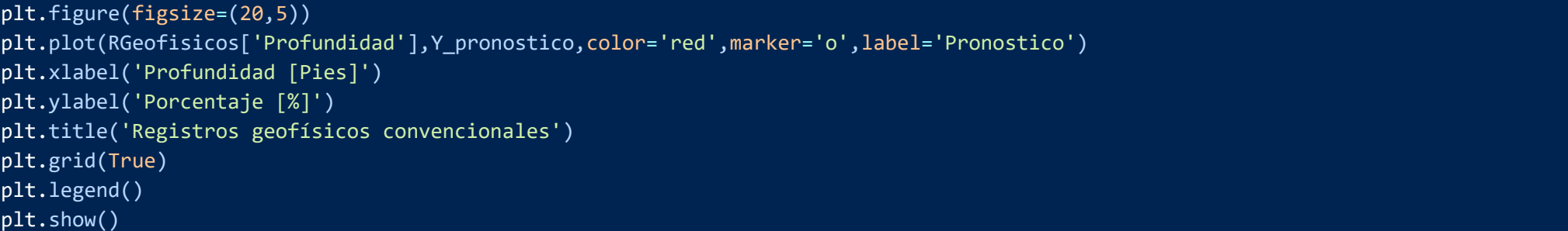
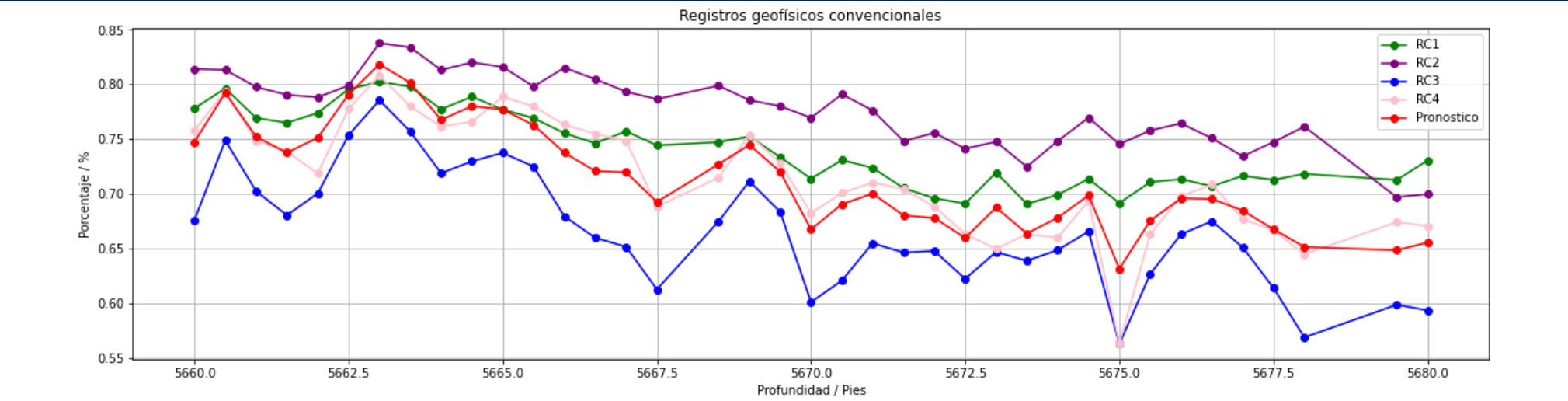
Comparación del modelo de pronóstico

$Y = a + b_1X_1 + b_2X_2 ... + b_nX_n + u$

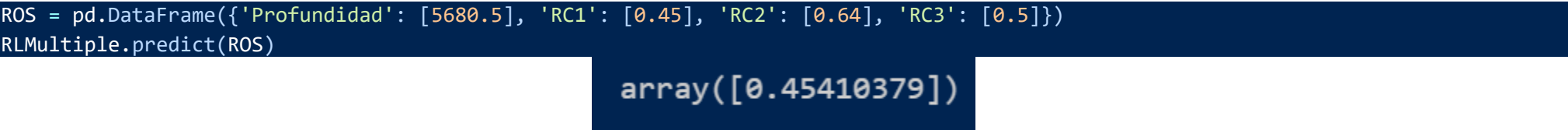
$Y = 0.2624 - 0.000075(Profundidad) + 0.5066(RC1) + 0.2275(RC2) + 0.4891(RC3) + 0.0684$

- Se tiene un **Score 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.

Proyección de los valores reales y pronosticados



Nuevos pronósticos



En este caso, al dar estos nuevos valores y mediciones, se obtiene un Score de 0.4541, es decir, un 45.41% de efectividad.

Conclusiones

A lo largo de esta práctica, a través de cuatro mediciones de los registros geofísicos convencionales, los cuales son: (RC1) Registro Neutrón, (RC2) Registro Sónico, (RC3) Registro Densidad-Neutrón, y (RC4) Registro Densidad Corregido por Arcilla, se obtuvo el pronóstico de la saturación de aceite remanente (ROS, Residual Oil Saturation), esto gracias a la aplicación del algoritmo de regresión lineal múltiple (ya que se tienen más de dos variables independientes), que pertenece a la categoría de aprendizaje supervisado, el cual su principal objetivo es predecir valores desconocidos o faltantes de una función de valor continuo.

Como se mencionó anteriormente, al aplicar este algoritmo, se obtuvo un **Score 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).

Por ende el modelo de pronóstico quedó de la siguiente manera:

$$Y = a + b_1X_1 + b_2X_2 \dots + b_nX_n + u$$

$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + u$$

Y: pronóstico de la saturación de aceite remanente (ROS, Residual Oil Saturation)

a: *intercepto*

b₁: *pendiente 1*

b₂: *pendiente 2*

b₃: *pendiente 3*

b₄: *pendiente 4*

u: *Residuo*

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

En adición a esto los pronósticos del modelo final se alejan en promedio **0.0004** y **0.0195** unidades del valor real, esto es, **MSE** y **RMSE**, respectivamente.