

Basile Álvarez Andrés José

No. Cuenta: 316617187

Email: andresbasile123@gmail.com

Fecha: 11/10/21

Inteligencia Artificial

Grupo III

Semestre 2022-1

Reporte Práctica 8: Pronóstico con regresión lineal múltiple

Objetivo: Obtener un pronóstico para datos de cáncer de mama en pacientes utilizando la regresión lineal múltiple.

Fuente de datos: Estudios clínicos a partir de imágenes digitalizadas de pacientes con cáncer de mama de Wisconsin (WDBC, Wisconsin Diagnostic Breast Cancer), donde:

- ID number: Identifica al paciente (valor discreto).
- Diagnosis: Diagnóstico (M=maligno, B=benigno).
- Radius: Media de las distancias del centro y puntos del perímetro.
- Texture: Desviación estándar de la escala de grises.
- Perimeter: Valor del perímetro del cáncer de mama.
- Area: Valor del área del cáncer de mama.
- Smoothness: Variación de la longitud del radio.
- Compactness: $\text{Perímetro}^2 / \text{Área} - 1$
- Concavity: Caída o gravedad de las curvas de nivel.
- Concave Points: Número de sectores de contorno cóncavo.
- Symmetry: Simetría de la imagen
- Fractal dimensión: Aproximación de frontera – 1.

[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

Características Generales:

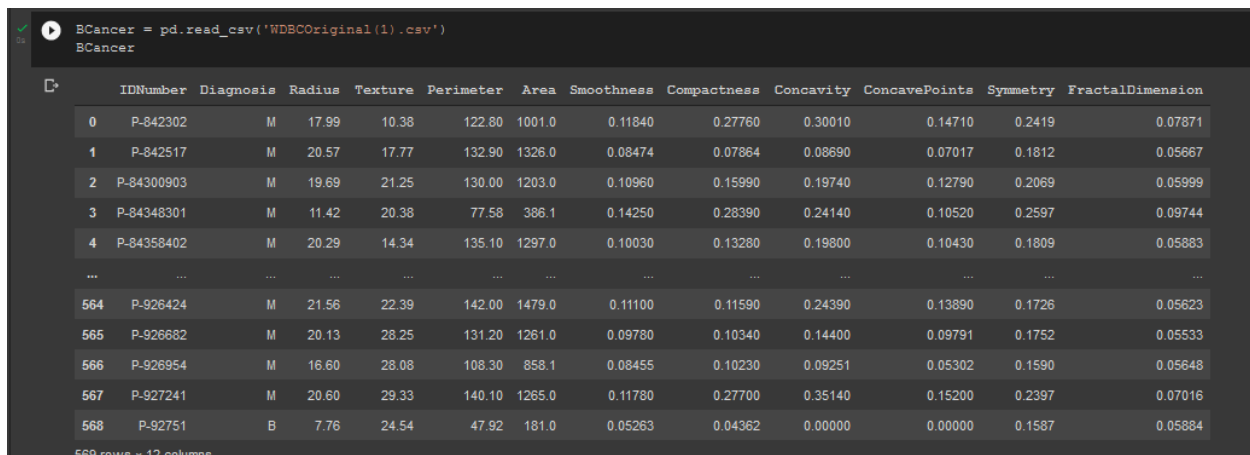
En esta práctica, utilizaremos la regresión lineal múltiple (trabajando con variables relacionadas a tumores de mama en pacientes). A lo largo del algoritmo, utilizaremos únicamente algunas variables del conjunto de variables totales para que funjan como variables predictoras y otra (área) para que sea la variable a predecir.

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 *WDBCOriginal.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:



```
Bcancer = pd.read_csv('WDBCOriginal(1).csv')
Bcancer
```

	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension
0	P-842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871
1	P-842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667
2	P-84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999
3	P-84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744
4	P-84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883
...
564	P-926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623
565	P-926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533
566	P-926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05648
567	P-927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016
568	P-92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587	0.05884

569 rows x 12 columns

Figura 1: Data Frame con algunos de los datos de tumores de mama.

Una vez hecho lo anterior, hacemos una gráfica de los registros de área del tumor por paciente en nuestro conjunto de datos, colocando en el eje horizontal al paciente y en el vertical el tamaño del tumor, mostrando los puntos que representan al área de cada uno de los tumores. Estos puntos nos muestran que las mediciones del tumor van en promedio entre 300 aproximadamente y 2500. Dichos datos nos van a servir para hacer una separación de los datos y poder predecir el comportamiento del área de manera adecuada.

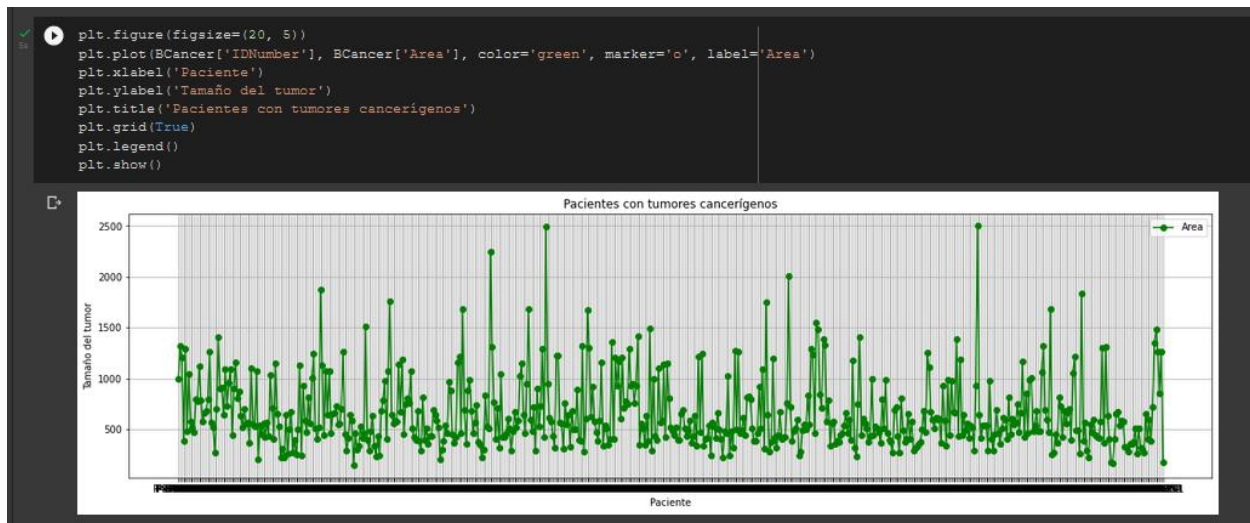


Figura 2: Gráfica de los registros de área de los tumores de mama.

Antes de aplicar el algoritmo de regresión lineal, utilizamos la matriz de correlaciones para conocer la dependencia que existe entre las variables y así poder realizar una selección de características que decremente la dimensionalidad total del conjunto de datos.

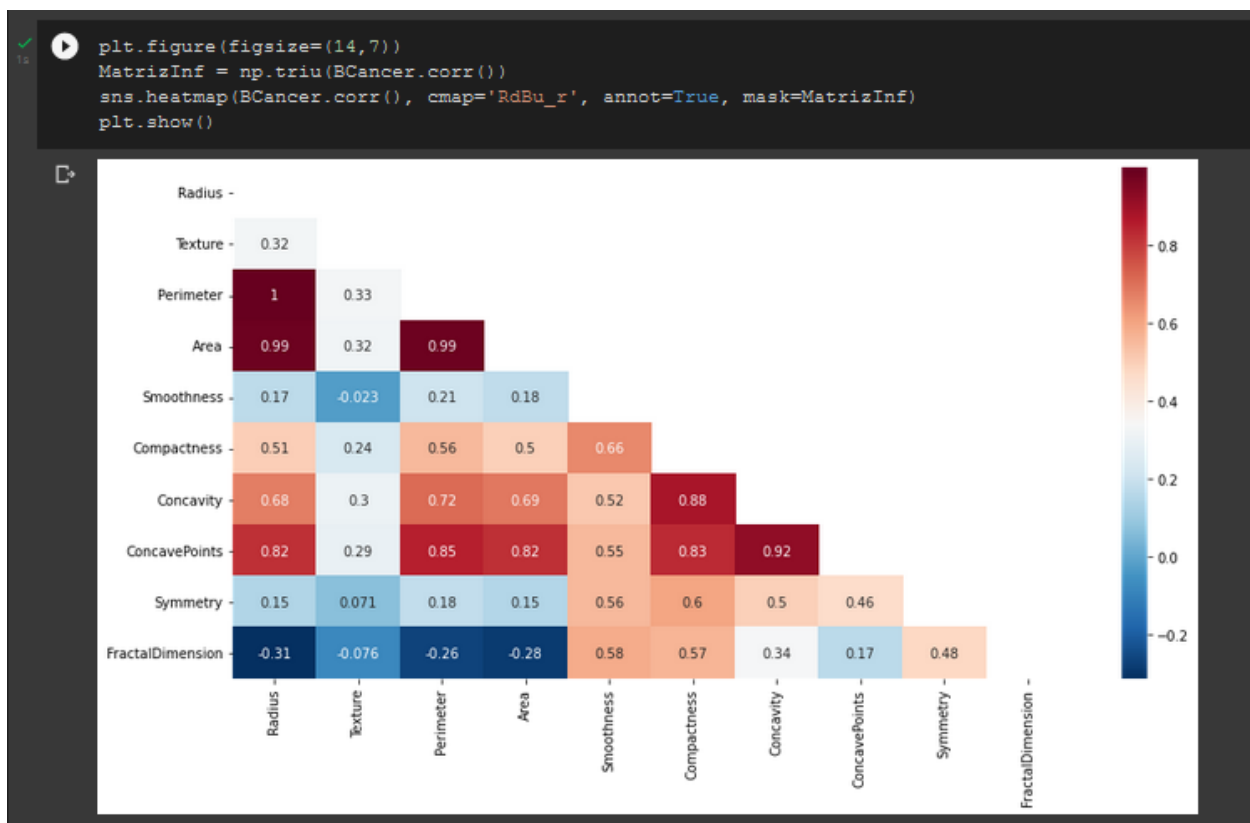


Figura 3: Matriz con las correlaciones entre variables.

Una vez hecho lo anterior, se seleccionan las variables: Texture, Area, Smoothness, Compactness, Symmetry, Fractal Dimension y Perimeter.

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, hacemos una matriz con *numpy* en donde colocamos las variables Texture, Perimeter, Smoothness, Compactness, Symmetry, FractalDimension.

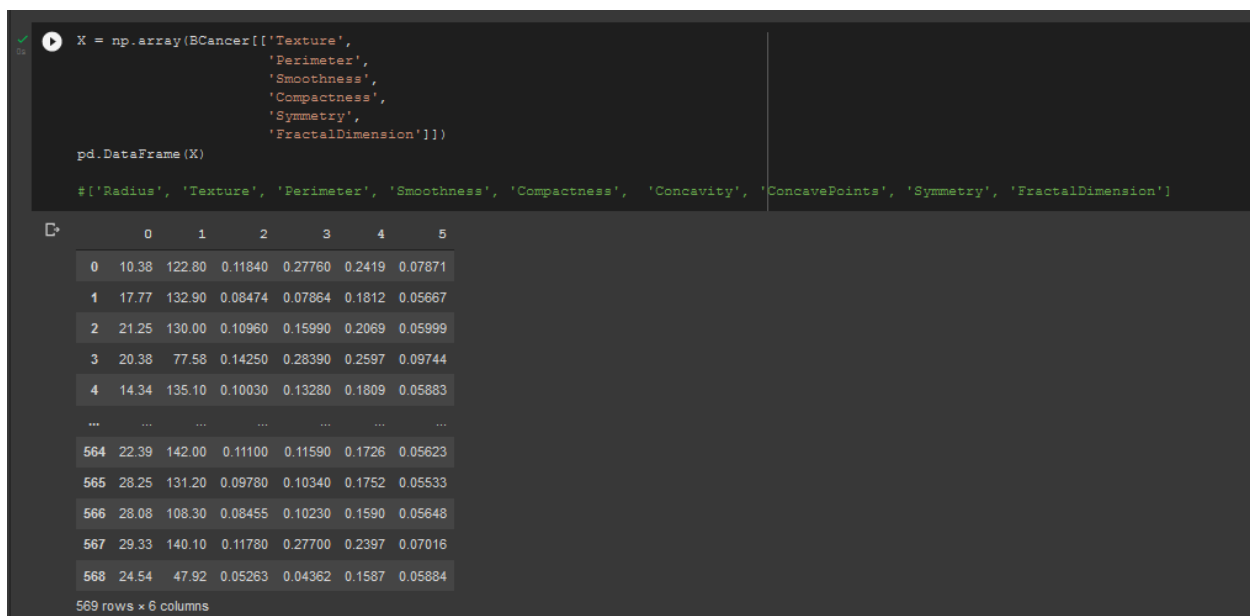


Figura 4: Parte de la matriz con las variables predictoras.

Para la variable a predecir, utilizaremos el Area, la cual colocamos en una matriz diferente, como se muestra en la figura 5.

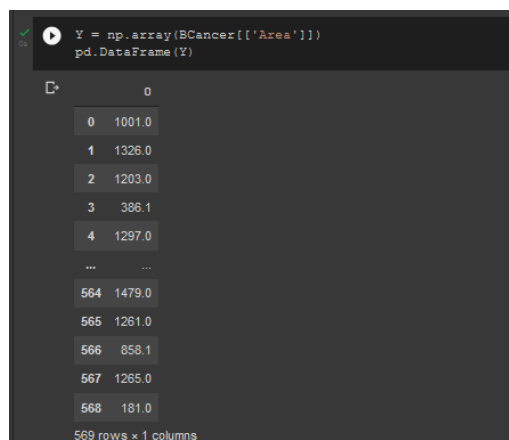


Figura 5: Parte de la matriz con la variable a predecir.

Se crean pequeños objetos donde se encontrarán los datos, tanto para X como para Y. Utilizando el *model_selection*, entrenamos y validamos los datos, dejando un porcentaje para la prueba (20% del global para la prueba y 80% para el entrenamiento), con un *random_state* específico para poder reproducir los mismos errores en caso de que ocurran. El aprendizaje del algoritmo, una vez aplicado, será a partir de los datos de entrenamiento dados.

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, y luego de hacer la división de los datos utilizando el método *model_selection.train_test_split*, generamos el pronóstico para los valores de *X_test* (variables predictoras nuevas que no fueron utilizadas en el entrenamiento) y colocamos el resultado en un *DataFrame*, como se muestra en la figura 6.

```
[19] #Se genera el pronóstico
Y_Pronostico = RLMultiple.predict(X_test)
pd.DataFrame(Y_Pronostico)
```

	0
0	405.607887
1	334.291077
2	505.762398
3	207.726058
4	604.229256
...	...
109	394.439214
110	1107.202694
111	541.131191
112	570.702628
113	2044.635054

114 rows x 1 columns

Figura 6: Parte del 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 tuvo un valor de 456.3649). Luego, calculamos el *Score* o bondad de ajuste, donde obtuvimos un valor de 0.9769 que indica que se tuvo una bondad de ajuste del 97%, la cual es considerada buena, ya que el pronóstico del área del tumor se logrará con un 97% de efectividad. Los valores obtenidos se presentan en la figura 7.

```

print('Coeficientes: \n', RLMultiple.coef_)
print('Intercepto: \n', RLMultiple.intercept_)
print("Residuo: %.4f" % max_error(Y_test, Y_Pronostico))
print("MSE: %.4f" % mean_squared_error(Y_test, Y_Pronostico))
print("RMSE: %.4f" % mean_squared_error(Y_test, Y_Pronostico, squared=False))
print('Score (Bondad de ajuste): %.4f' % r2_score(Y_test, Y_Pronostico))

Coeficientes:
[[ 6.86261446e-01  1.63885604e+01  2.50787388e+01 -1.40602548e+03
   1.46803422e+02  6.23269303e+03]]
Intercepto:
[-1140.33616115]
Residuo: 456.3649
MSE: 3083.2634
RMSE: 55.5271
Score (Bondad de ajuste): 0.9769

```

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

Posteriormente, obtuvimos el modelo matemático para el ejemplo utilizado. Esto es equivalente a obtener la recta (en este caso, por ser de múltiples variables, se genera una especie de hiperplano) que minimiza la distancia entre la línea ajustada y los puntos de los datos. Para el modelo de pronóstico, obtenemos:

$$Y = -1140.34 + 0.69(\text{Texture}) + 16.39(\text{Perimeter}) + 25.08(\text{Smoothness}) - 1406.03(\text{Compactness}) + 146.80(\text{Symmetry}) + 6232.69(\text{FractalDimension}) + 456.36$$

- Se tiene un Score de 0.9769, el cual indica que el pronóstico del Área del tumor se logrará con un 97.69% de efectividad.
- Además, los pronósticos del modelo final se alejan en promedio 3083.26 y 55.53 unidades del valor real, esto es, MSE y RMSE, respectivamente.

Figura 8: Conformación del modelo de pronóstico.

Más tarde, obtuvimos el modelo matemático para el ejemplo utilizado, pero ahora con todas las variables (sin tomar en cuenta la selección que habíamos realizado anteriormente). Para el modelo de pronóstico, obtenemos:

$$Y = -1140.34 + 0.69(\text{Texture}) + 16.39(\text{Perimeter}) + 25.08(\text{Smoothness}) - 1406.03(\text{Compactness}) + 146.80(\text{Symmetry}) + 6232.69(\text{FractalDimension}) + 456.36$$

- Se tiene un Score de 0.9769, el cual indica que el pronóstico del Área del tumor se logrará con un 97.69% de efectividad.
- Además, los pronósticos del modelo final se alejan en promedio 3083.26 y 55.53 unidades del valor real, esto es, MSE y RMSE, respectivamente.

Figura 9: Conformación del modelo de pronóstico con todas las variables.

Finalmente, podemos obtener pronósticos para la variable *Area* a partir del modelo generado, ingresando los valores de las variables *Texture*, *Perimeter*, *Smoothness*, *Compactness*, *Symmetry*, y *FractalDimension*. Esto nos será útil para encontrar la correspondencia del área del tumor 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 médica (área de un tumor de mama) a partir de una regresión lineal múltiple con 6 variables predictoras. Esto es un algoritmo de aprendizaje supervisado.

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

Este tipo de algoritmo es muy utilizado en diversas aplicaciones. Si bien en esta práctica utilizamos este tipo de aprendizaje para datos médicos, esto se podría traducir a datos de todo tipo como para obtener valores pronosticados de temperatura, predecir el clima o incluso para verificar si el solicitante a un trabajo cumple con los criterios mínimos para el puesto, 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.