Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 16/04/2022
Practica No.9 Regresión Lineal Múltiple.

Objetivo.

Obtener grupos de pacientes con características similares, diagnosticadas con un tumor de mama, a través de clustering jerárquico y particional.

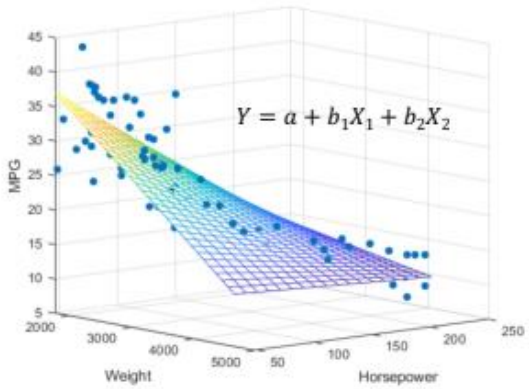
Características.

Estudios clínicos a partir de imágenes digitalizadas de pacientes con cáncer de mama de Wisconsin (WDBC, Wisconsin Diagnostic Breast Cancer).

La regresión lineal múltiple es un algoritmo que busca calcular una salida Y (Una variable dependiente) en función de múltiples entradas (Variables dependientes) calculando la ecuación que genere la recta o el hiperplano el cual minimiza la distancia entre cada uno de los puntos y el hiperplano ajustado.

$$\hat{Y} = a + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n + \mu$$

Donde μ, es el residuo del modelo que representa la diferencia entre el punto pronosticado y la observación real. Es importante considerar que al ser un algoritmo de aprendizaje supervisado los datos tienen que estar etiquetados, es decir, que todos deberán tener un valor real de la variable dependiente el cual se va a utilizar para entrenar al modelo.



Desarrollo.

Como primer paso tenemos la importación de las librerías necesarias para trabajar con el conjunto de datos de los pacientes a segmentar. `pandas` para la manipulación de los datos, `numpy` para el manejo de matrices, `matplotlib` y `seaborn` para la visualización de estos datos mediante gráficos de diferente tipo dependiendo del análisis.

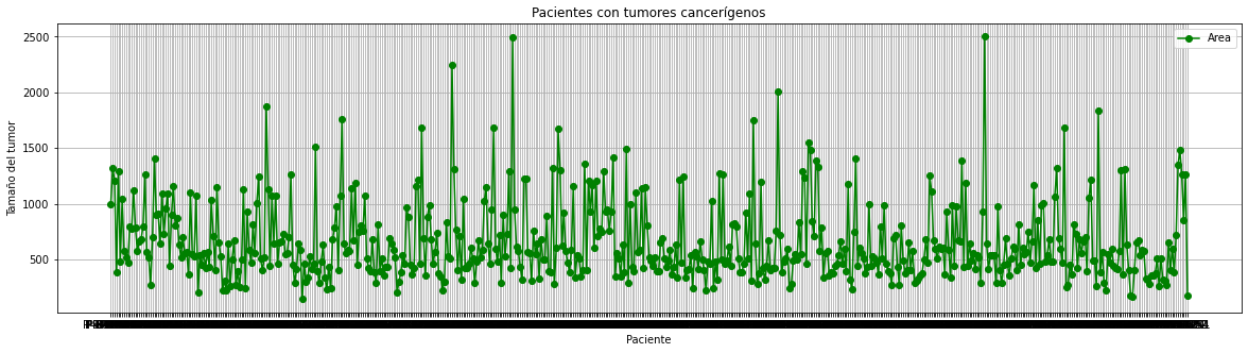
Después tenemos que hacer la lectura de nuestros datos los cuales están relacionados a las características de los tumores presentados en cada uno de los pacientes relacionadas como su área, textura y la etiqueta la cual nos indica si se trata de un tumor maligno o benigno que nos ayudara mas adelante para el proceso de entrenamiento del modelo.

Tenemos que leer los datos con ayuda de pandas el cual nos genera un `dataframe` con 569 registros y 12 columnas asociadas a las características de los tumores presentados.

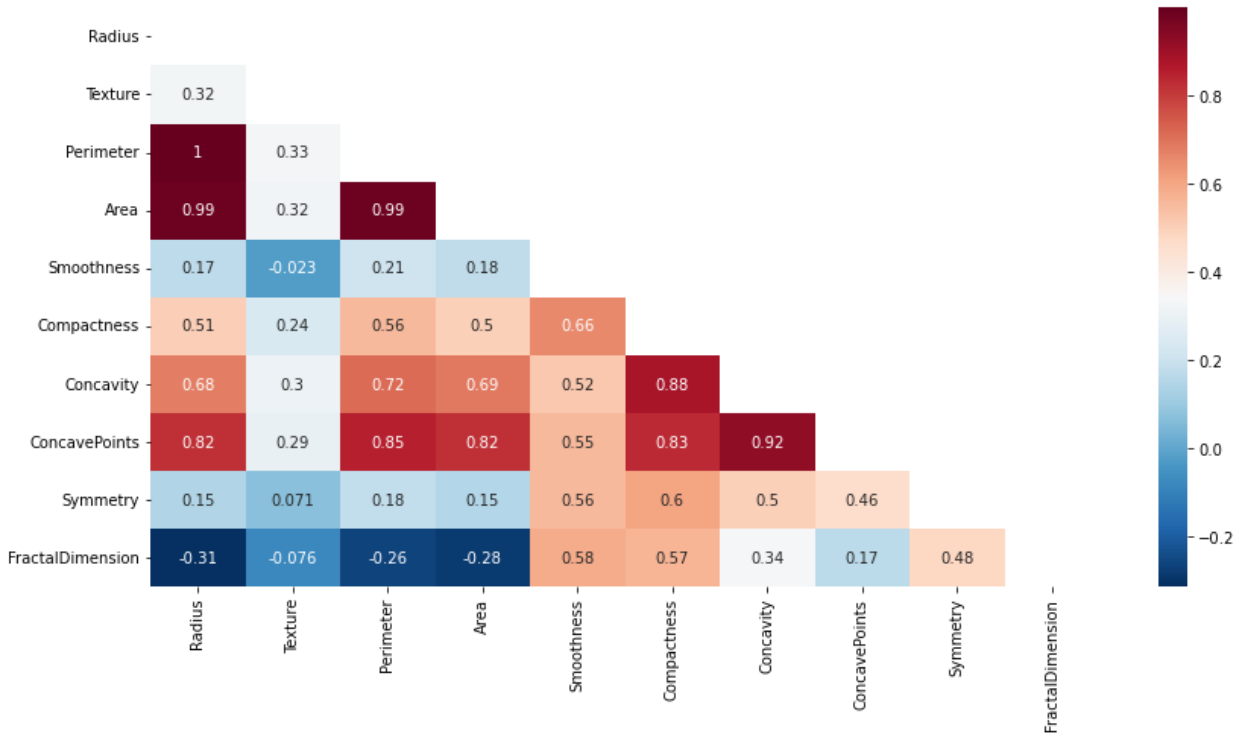
El conjunto de datos leídos se muestra a continuación.

	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

A continuación, se muestra la gráfica en la cual podemos ver para cada uno de los pacientes cual es el tamaño aproximado del tumor ya que esta es una característica que generalmente indica si es un tumor maligno o benigno.



Para las 12 variables en el modelo se decidió hacer un análisis para la selección de características trazando para ello un mapa de calor el cual nos muestra con un índice de -1 a 1 el nivel de correlación que tiene cada variable con las demás. A continuación, se muestra el mapa de calor obtenido resaltando las relaciones fuertes entre área y perímetro, radio y área, Concavity con la mayoría de las variables, Concave points de igual manera con la mayoría de las variables por lo cual se decidió que se eliminarían del modelo.



Al final del análisis de correlación las variables que se seleccionaron fueron las siguientes.

1.- Texture.

2.- Area.

3.- Smoothness.

4.- Compactness.

5.- Symmetry.

6.- FractalDimension.

Ya con cada una de las variables del modelo definidas podemos comenzar a crearlo, para ello nos ayudaremos de las librerías de `linear_model` para generar el modelo, `mean_squared_error`, `max_error` y `r2_score` los cuales nos van a ayudar a generar las métricas para evaluar la efectividad del modelo una vez lo tengamos.

Primero se tiene que seleccionar la variable dependiente del modelo (X) la cual será el área del tumor y cuáles serán las variables independientes (Y); Texture, Area, Smoothness, Compactness, Symmetry y FractalDimension. Con esto podemos hacer 2 arreglos de `numpy` a los cuales llamaremos `X` y `Y`, sin embargo, aún falta el separar el conjunto de 559 registros en uno de entrenamiento y otro para las pruebas, para ello se usó la librería de `sklearn model_selection` con su método `train_test_split` el cual recibe como datos el arreglo de `numpy` con las variables independientes del modelo (X), el arreglo de `numpy` con las variables dependientes (Y) o etiquetas, cuál será el tamaño en porcentaje del conjunto de

pruebas, una semilla y el atributo `shuffle` el cual mezcla los datos de tal forma que no se tomen según la secuencia dada en un inicio. Como resultado ahora se tienen los 4 conjuntos 2 para entrenamiento y 2 para las pruebas con una proporción de 80% (455 registros) para el entrenamiento y 20% (114 registros) para las pruebas.

```
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(X, Y,
                                                                    test_size = 0.2,
                                                                    random_state = 1234,
                                                                    shuffle = True)
```

Después de la separación del conjunto de datos se procede con la creación del modelo generado como un objeto de la clase `linear_model.LinearRegression` (Al cual llamamos RLMultiple) podemos utilizar el método `fit` pasando como datos los parámetros de los arreglos con los conjuntos de entrenamiento. Después de esto podemos utilizar el modelo de regresión lineal múltiple generado a partir de estos datos con el método `predict`.

```
RLMultiple = linear_model.LinearRegression()
RLMultiple.fit(X_train, Y_train)
……
Y_pronostico = RLMultiple.predict(X_train)
pd.DataFrame(Y_pronostico)
```

Esto nos generara los valores que se pronostican para cada uno de los registros con los que se entrenó el modelo, sin embargo, debemos recordar que estos valores presentan cierto grado de error al no ajustarse totalmente al hiperplano.

Con los atributos `coef_` e `intercept_` del modelo (RLMultiple) podemos consultar cuales son los coeficientes de nuestro hiperplano el cual se va a ajustar a la mayoría de los registros siguiendo la siguiente ecuación.

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

Sin embargo, aun faltaría agregar el residuo del modelo el cual lo podemos obtener con ayuda de la clase `max_error` que se encuentra en el paquete `sklearn.metrics` de Sklearn.

A continuación, se muestra el modelo de regresión lineal múltiple utilizado para hacer las predicciones.

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

Donde 456.36 representa el residuo, es decir una medida general la cual nos dice que tan diferentes son los valores pronosticados de los valores reales.

Además, podemos calcular el MSE y su raíz, el RMSE las cuales son dos métricas que nos indican en promedio que tantas unidades se alejan los valores pronosticados de los valores reales los cuales son 3086.26 y 55.53 respectivamente. Estos valores se calcularon con la clase `mean_squared_error` de Sklearn.

Como ultima métrica se calculó el score que indica el porcentaje de precisión del modelo siendo un indicativo de su efectividad con nuevos valores el cual se logró de 97.69% que para el contexto en el que se está trabajando resulta ser un modelo muy efectivo.

Por último, es importante tener en cuenta que con este modelo podemos llegar a hacer extrapolaciones ingresando nuevos valores para cada una de las variables independientes que seleccionamos para el modelo en un inicio, los cuales en este caso serían las características con las que presenta el paciente. Para probar nuestro modelo ingresamos datos para una textura de 10.38, un perímetro de 122.8, una suavidad de 0.11, una compacidad de 0.27, una simetría de 0.24 y una dimensión fractal de 0.08 obteniendo área del tumor aproximada a los 1018.04 unidades la cual no se aleja mucho del valor real para estas características el cual es 1001 unidades.

Modelo con todas las variables.

En esta ocasión se entreno al modelo con todas las variables numéricas del dataset disponibles las cuales son radiusm textura, perimeter, smoothness, compactness, concavity, concave points, symmetry y fractal dimensión (9 en total).

```
RLMultiple_all = linear_model.LinearRegression()
RLMultiple_all.fit(X_all_train, Y_all_train)
-----------------------------------------------------------
Y_all_Pronostico = RLMultiple_all.predict(X_all_test)
pd.DataFrame(Y_all_Pronostico)
```

El proceso para generar el modelo fue exactamente el mismo, solamente cambiaron los arreglos de numpy con los que se iba a generar ya que estos ahora tendrían que contemplar todas las variables que se mencionaron anteriormente. A continuación, se muestra la ecuación de la nueva regresión lineal múltiple.

$$\hat{Y} = -976.18 - 35.08(Radius) + 0.48(Texture) + 20.79(Perimeter) - 169.89(Smoothness) - 1894.45(Compactness) + 232.74(Concavity) + 529.22(ConcavePoints) + 66.61(Symmetry) + 5716.43(FractalDimension) + 425.23$$

En comparación con el modelo anterior podemos ver que el residuo redujo de 456 a 425 lo cual podría ser un indicativo que bajo este modelo se tendrá un mejor score.

El MSE y su raíz, el RMSE las cuales son dos métricas que nos indican en promedio que tantas unidades se alejan los valores pronosticados de los valores reales y pare este modelo tienen un valor de 2932.75 y 54.15 respectivamente

Por último, el score que indica el porcentaje de precisión del modelo siendo un indicativo de su efectividad con nuevos valores el cual se logró de 97.8% incrementando 0.11% en comparación con el anterior obteniendo una mejor métrica bajo la condición de que generar el modelo requerirá de mayor costo computacional al tener que contemplar un mayor número de variables.

En la extrapolación con este modelo podemos ver como es que se comportan con datos reales teniendo que para un registro con características que resultan en un área de 1001 unidades el modelo con reducción de variables nos entregó un área de 1018.04 y el que contempla todas las variables un área de 1019.42 el cual resulta ser un peor valor en comparación con el que contempla la reducción de variables. Con esto podemos decir que si bien contemplar un mayor numero de variables en el modelo por lo general entregara modelos un poco mas precisos, este incremento puede no ser lo suficientemente significativo para el análisis además de que requiere un mayor costo computacional.

Conclusiones.

En esta práctica se logró generar un modelo de regresión lineal múltiple el cual ajusta un hiperplano a una serie de registros. Dicho modelo predice con un valor continuo cual es el área de un tumor en cierto paciente que presenta una serie de características, para ello se emplearon 2 estrategias. La primera de ellas consistió en hacer una reducción de variables para pasar de 9 a 6 variables contempladas como variables independientes del modelo mientras que en la segunda se contemplaron las 9 variables, se observo un incremento en el score y un decremento en el residuo del modelo que contemplo todas las variables, sin embargo, este incremento no fue lo suficientemente significativo y el costo computacional se incrementó.

Se obtuvieron diferentes métricas para ambos modelos el residuo (456.36 y 425.22) que es directamente la diferencia entre el valor pronosticado y el valor real generado un solo índice considerando cada registro. El error cuadrático medio y su raíz, MSE (3083.26 y 2832.75) y RMSE (55.52 y 54.15) respectivamente que indican las unidades en promedio que las predicciones se alejan del valor real y por último el Score del modelo que nos va a representar con un porcentaje que tan preciso resulto ser el modelo obteniendo un 97.69% y 97.8% que bajo este contexto en una métrica más que aceptable.

Fuentes.

pandas documentation — pandas 1.4.1 documentation. (2022). Pandas. Recuperado 2022, de https://pandas.pydata.org/docs/index.html#

User guide: contents. (2022). Scikit-Learn. Recuperado 2022, de https://scikit-learn.org/stable/user_guide.html