Universidad Nacional Autónoma de México
Facultad de Ingeniería

Pronóstico con regresión lineal múltiple

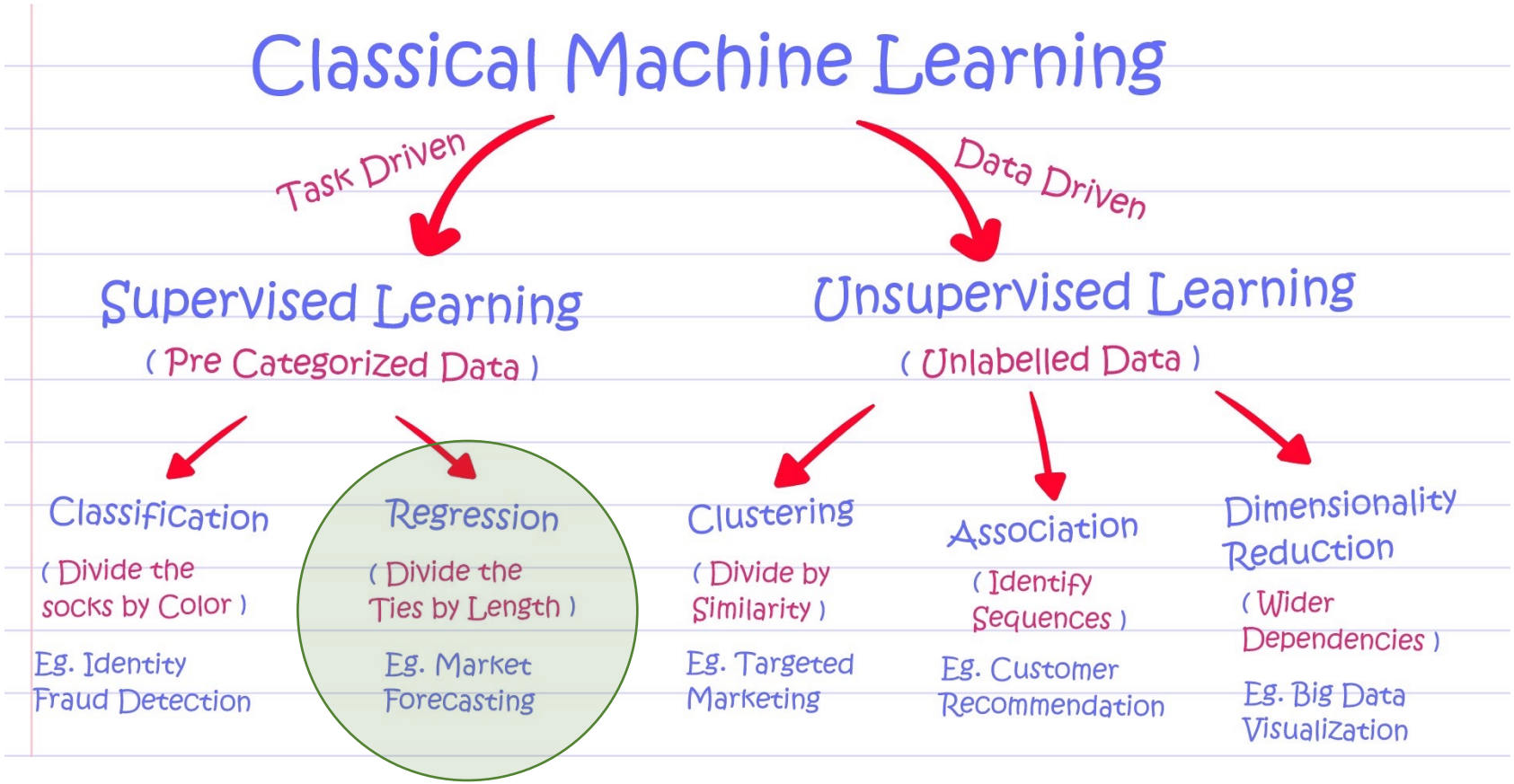
Enfoque de aprendizaje supervisado

Práctica 8

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

Noviembre, 2021

Práctica


Classical Machine Learning
Task Driven
Data Driven
Supervised Learning
(Pre Categorized Data)
Unsupervised Learning
(Unlabelled Data)
Classification
(Divide the socks by Color)
Eg. Identity Fraud Detection
Regression
(Divide the Ties by Length)
Eg. Market Forecasting
Clustering
(Divide by Similarity)
Eg. Targeted Marketing
Association
(Identify Sequences)
Eg. Customer Recommendation
Dimensionality Reduction
(Wider Dependencies)
Eg. Big Data Visualization

Práctica

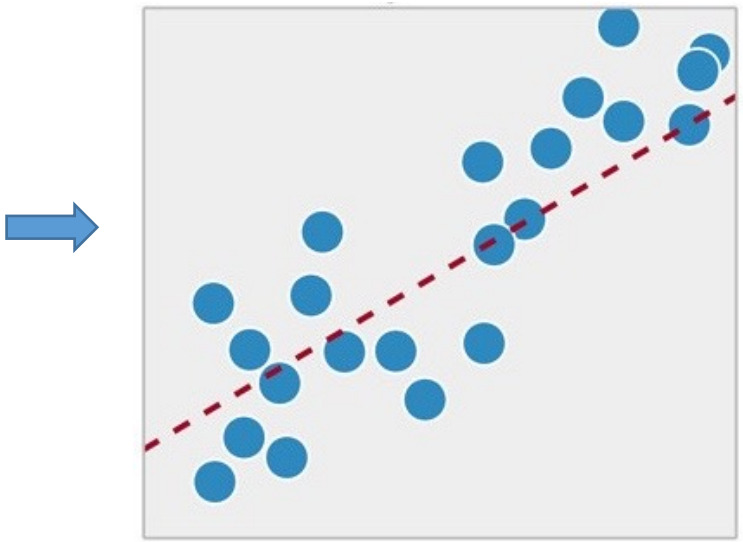
HISTORICAL DATA

TIME		FEATURE 1 (X1)	FEATURE 2 (X2)	FEATURE 3 (X3)	.....	TARGET (Y)
↓	T1	.	.	.		Y1
	T2	.	.	.		Y2
	T3	.	.	.		Y3

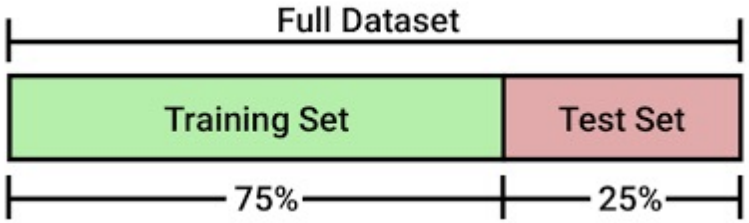
	(X1)	(X2)	(X3)	.....
T10	.	.	.	
T11	.	.	.	

X -> ML / MODEL -> Y

PREDICTED Y



Pronóstico



Práctica

Fuente de datos

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

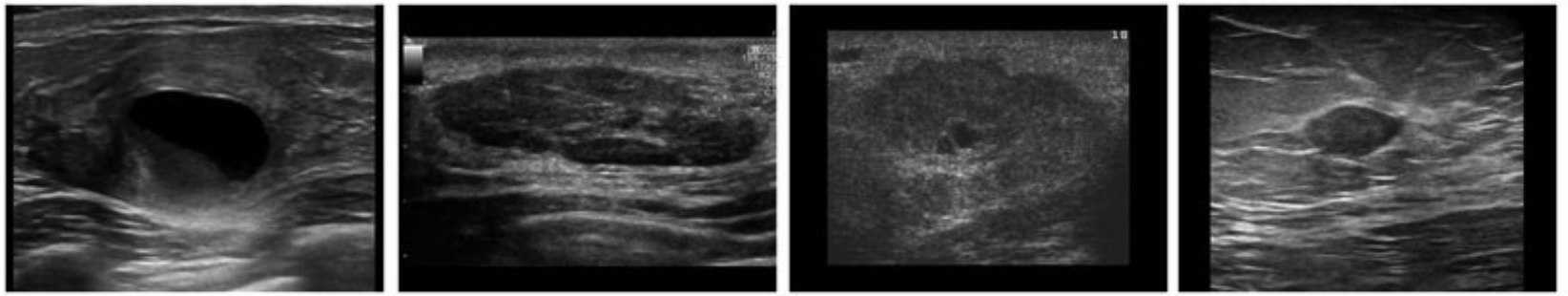
Variable	Descripción	Tipo
ID number	Identifica al paciente	Discreto
Diagnosis	Diagnostico (M=maligno, B=benigno)	Booleano
Radius	Media de las distancias del centro y puntos del perímetro	Continuo
Texture	Desviación estándar de la escala de grises	Continuo
Perimeter	Valor del perímetro del cáncer de mama	Continuo
Area	Valor del área del cáncer de mama	Continuo
Smoothness	Variación de la longitud del radio	Continuo
Compactness	Perímetro ^ 2 /Area - 1	Continuo
Concavity	Caída o gravedad de las curvas de nivel	Continuo
Concave points	Número de sectores de contorno cóncavo	Continuo
Symmetry	Simetría de la imagen	Continuo
Fractal dimension	“Aproximación de frontera” - 1	Continuo

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

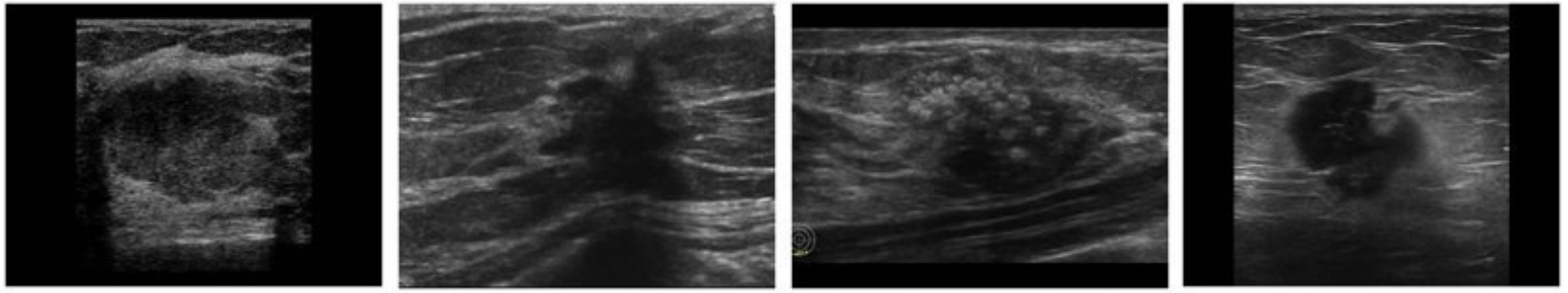
Fuente de datos

Registros clínicos de cáncer de mama a partir de imágenes digitalizadas.

Benigno



Maligno



Práctica

1. Importar las bibliotecas y los datos

```
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
```

Práctica

1. Importar las bibliotecas y los datos

```
BCancer = pd.read_csv('WDBCOriginal.csv')
BCancer
```

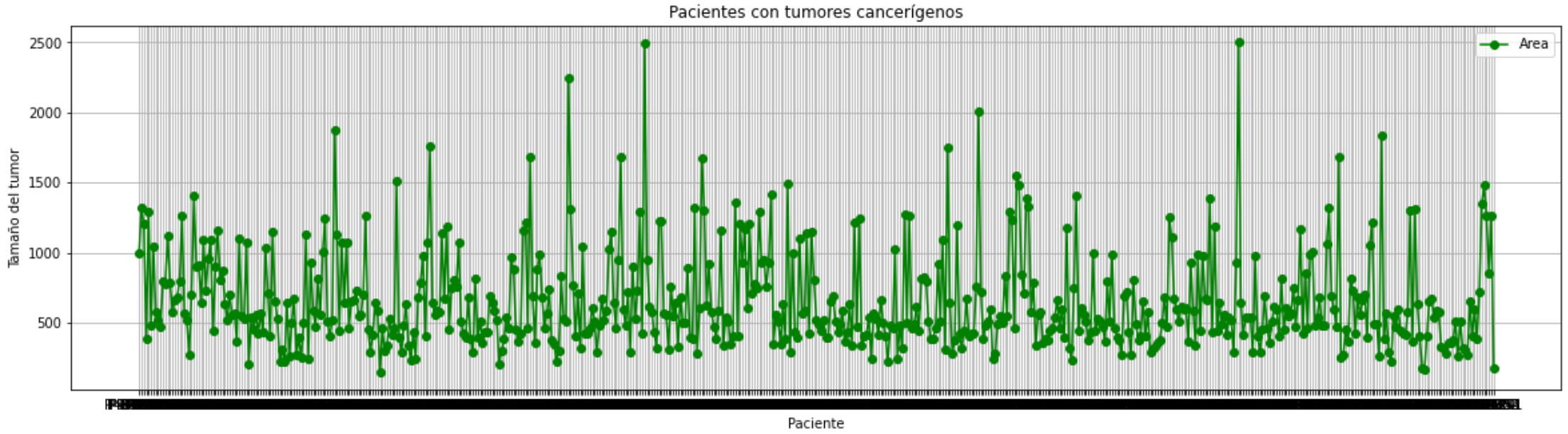
	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension
0	P-842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871
1	P-842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667
2	P-84300903	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 × 12 columns

Práctica

2) Gráfica del área del tumor por paciente

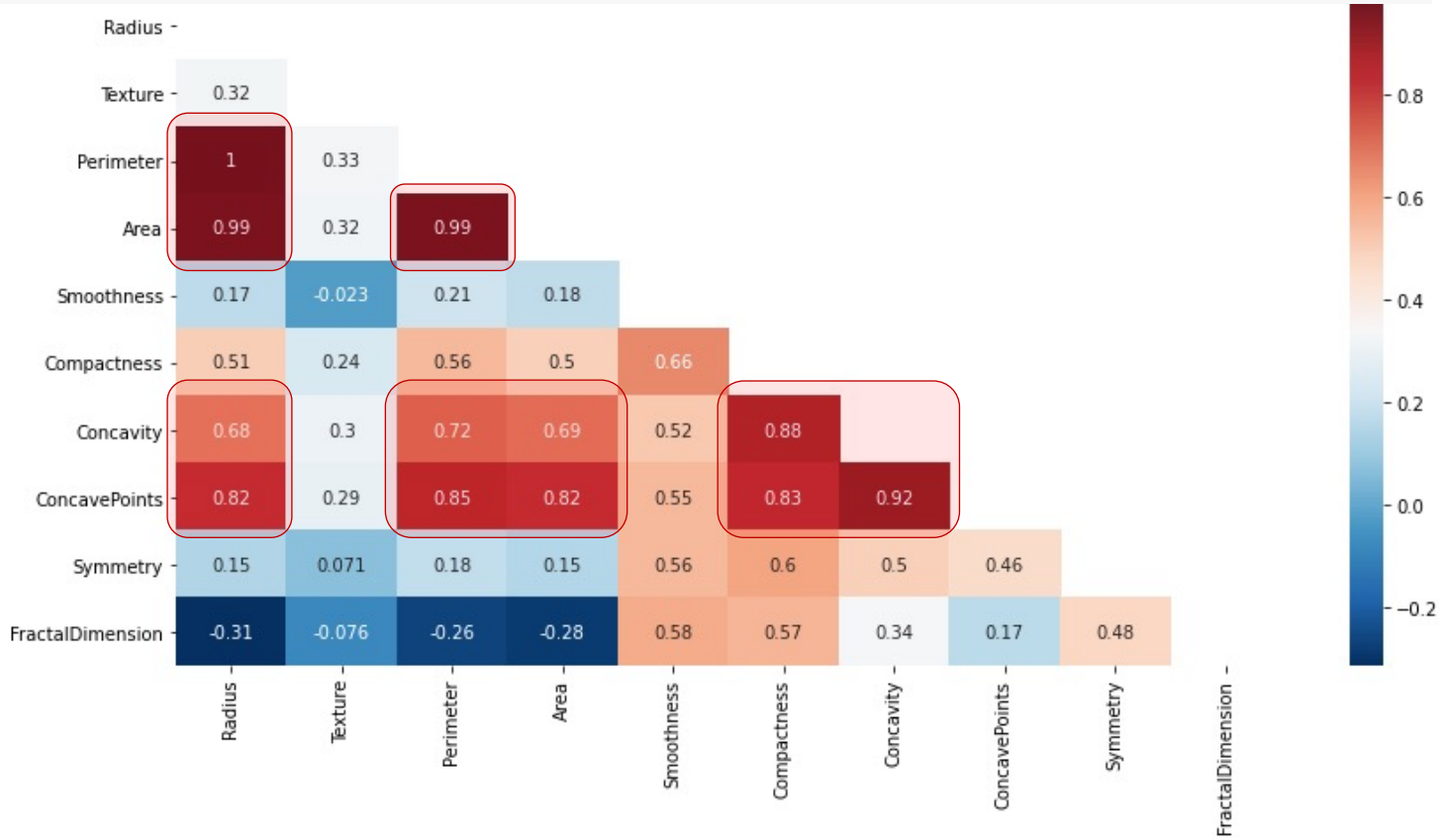
```
plt.figure(figsize=(20, 5))
plt.plot(BCancer['IDNumber'], BCancer['Area'], color='green', marker='o', label='Area')
plt.xlabel('Paciente')
plt.ylabel('Tamaño del tumor')
plt.title('Pacientes con tumores cancerígenos')
plt.grid(True)
plt.legend()
plt.show()
```



Práctica

3. Selección de características

```
plt.figure(figsize=(14,7))
MatrizInf = np.triu(BCancer.corr())
sns.heatmap(BCancer.corr(), cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```



Varibles seleccionadas:

1) Textura [Posición 3]
2) Area [Posición 5]
3) Smoothness [Pos. 6]
4) Compactness [Pos. 7]
5) Symmetry [Posición 10]
6) FractalDimension [Pos. 11]
*7) Perimeter [Posición 4] - Para calcular el área del tumor -

4. Aplicación del algoritmo

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

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

```
X = np.array(BCancer[['Texture',
                      'Perimeter',
                      'Smoothness',
                      'Compactness',
                      'Symmetry',
                      'FractalDimension']])
pd.DataFrame(X)
```

```
Y = np.array(BCancer[['Area']])
pd.DataFrame(Y)
```

4. Aplicación del algoritmo

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

```
pd.DataFrame(X)
```

	0	1	2	3	4	5
0	10.38	122.80	0.11840	0.27760	0.2419	0.07871
1	17.77	132.90	0.08474	0.07864	0.1812	0.05667
2	21.25	130.00	0.10960	0.15990	0.2069	0.05999
3	20.38	77.58	0.14250	0.28390	0.2597	0.09744
4	14.34	135.10	0.10030	0.13280	0.1809	0.05883
...	...	...	...	...	...	...

```
pd.DataFrame(Y)
```

	0
0	1001.0
1	1326.0
2	1203.0
3	386.1
4	1297.0
...	...

4. Aplicación del algoritmo

Se hace la división de los datos

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

```
pd.DataFrame(X_train)
#pd.DataFrame(X_test)
```

	0	1	2	3	4	5
0	18.22	84.45	0.12180	0.16610	0.1709	0.07253
1	22.44	71.49	0.09566	0.08194	0.2030	0.06552
2	20.76	82.15	0.09933	0.12090	0.1735	0.07070
3	23.84	82.69	0.11220	0.12620	0.1905	0.06590
4	18.32	66.82	0.08142	0.04462	0.2372	0.05768
...	...	...	...	...	...	...

```
pd.DataFrame(Y_train)
#pd.DataFrame(Y_test)
```

	0
0	493.1
1	378.4
2	480.4
3	499.0
4	340.9
...	...

4. Aplicación del algoritmo

Se entrena el modelo a través de 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)
```

4. Aplicación del algoritmo

Se genera el pronóstico

```
#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 × 1 columns

5. Obtención de los coeficientes, intercepto, error y Score

```
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))   #True devuelve MSE, False devuelve RMSE
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
```

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

Y = −1140.34 + 0.69(Texture) + 16.39(Perimeter) + 25.08(Smoothness) − 1406.03(Compactness) + 146.80(Symmetry) + 6232.69(FractalDimension) + 456.36

6. Conformación del modelo de pronóstico

```
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
```

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

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

- Se tiene un Score de 0.9769, el cual indica que el pronóstico del Area 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.

7. Nuevos pronósticos

```
AreaTumor = pd.DataFrame({'Texture': [18.32],
                          'Perimeter': [66.82],
                          'Smoothness': [0.08142],
                          'Compactness': [0.04462],
                          'Symmetry': [0.2372],
                          'FractalDimension': [0.05768]})
RLMultiple.predict(AreaTumor)
```

```
array([[300.94831572]])
```