Contenido

[Contexto 2](#_Toc87356729)

[Objetivo 2](#_Toc87356730)

[Fuente de datos 2](#_Toc87356731)

[Preparación del entorno de ejecución 2](#_Toc87356732)

[1) Importar las bibliotecas necesarias 2](#_Toc87356733)

[2) Importar los datos 3](#_Toc87356734)

[Gráfica del área del tumor por paciente 3](#_Toc87356735)

[Selección de características 3](#_Toc87356736)

[Aplicación del algoritmo 4](#_Toc87356737)

[Se seleccionan las variables predictoras (X) y la variable a pronosticar (Y) 4](#_Toc87356738)

[Se hace la división de los datos 4](#_Toc87356739)

[Se entrena el modelo a través de Regresión Lineal Múltiple 5](#_Toc87356740)

[Se genera el pronóstico 5](#_Toc87356741)

[Obtención de los coeficientes, intercepto, error y Score 5](#_Toc87356742)

[Conformación del modelo de pronóstico 6](#_Toc87356743)

[a) Solo con las variables seleccionadas 6](#_Toc87356744)

[b) Tomando en cuenta todas las variables 6](#_Toc87356745)

[Nuevos pronósticos 6](#_Toc87356746)

[Conclusiones 6](#_Toc87356747)

# Contexto

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

Diagrama

Descripción generada automáticamente

## Fuente de datos

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

Texto

Descripción generada automáticamente

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

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

Escala de tiempo

Descripción generada automáticamente

# Preparación del entorno de ejecución

## 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

## Importar los datos

Fuente de datos: WDBCOriginal.csv

# Si se usa Google Colab

#from google.colab import files

#files.upload()

# Si se importan los datos desde Drive

#from google.colab import drive

#drive.mount('/content/drive')

BCancer = pd.read\_csv("WDBCOriginal.csv")

BCancer

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# 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()

Gráfico

Descripción generada automáticamente

# 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()

Gráfico

Descripción generada automáticamenteVariables seleccionadas:

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

# 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)

|  |  |
| --- | --- |
| **Código** | **Salida** |
| X = np.array(BCancer[['Texture',                        'Perimeter',                        'Smoothness',                        'Compactness',                        'Symmetry',                        'FractalDimension']])  pd.DataFrame(X)  # Tomando en cuenta todas las variables  #['Radius', 'Texture', 'Perimeter', 'Smoothness', 'Compactness',    'Concavity', 'ConcavePoints', 'Symmetry',   'FractalDimension'] | Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente |
| Y = np.array(BCancer[['Area']])  pd.DataFrame(Y) |  |

## 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)

# Se deja un espacio de 20% para la prueba y un 80% para el entrenamiento

|  |  |
| --- | --- |
| **Código** | **Salida** |
| pd.DataFrame(X\_train)  # pd.DataFrame(X\_test) | Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico  Descripción generada automáticamente |
| pd.DataFrame(Y\_train)  # pd.DataFrame(Y\_test) | Imagen de la pantalla de un celular con letras  Descripción generada automáticamente con confianza baja |

## Se entrena el modelo a través de Regresión Lineal Múltiple

RLMultiple = linear\_model.LinearRegression()

RLMultiple.fit(X\_train, Y\_train) #Se entrena el modelo

## Se genera el pronóstico

#Se genera el pronóstico

Y\_Pronostico = RLMultiple.predict(X\_test)

pd.DataFrame(Y\_Pronostico)

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

# 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))

# True devuelve MSE, False devuelve RMSE

print("RMSE: %.4f" % mean\_squared\_error(Y\_test, Y\_Pronostico, squared=False))

print('Score (Bondad de ajuste): %.4f' % r2\_score(Y\_test, Y\_Pronostico))

print('\n')

print("Pronóstico del área del Tumor: Y =",RLMultiple.intercept\_[0],"+", RLMultiple.coef\_[0][0], "(Texture) + ", RLMultiple.coef\_[0][1], "(Perimeter) + "

,RLMultiple.coef\_[0][2], "(Smoothness) + ", RLMultiple.coef\_[0][3], "(Compactness) + ",RLMultiple.coef\_[0][4], "(Symmetry) + "

,RLMultiple.coef\_[0][5], "(FractalDimension) + ", max\_error(Y\_test, Y\_Pronostico))

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# Conformación del modelo de pronóstico

## Solo con las variables seleccionadas

* 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.

## Tomando en cuenta todas las variables

* Se tiene un **Score de 0.9780**, el cual indica que el **pronóstico del Área del tumor se logrará con un 97.8% de efectividad.**
* Además, los pronósticos del modelo final se alejan en promedio 2932.75 y 54.15 unidades del valor real, esto es, MSE y RMSE, respectivamente.

# Nuevos pronósticos

AreaTumor = pd.DataFrame({'Texture': [18.32], 'Perimeter': [166.82], 'Smoothness': [0.08142],

                        'Compactness': [0.04462], 'Symmetry': [0.2372], 'FractalDimension': [0.05768]})

RLMultiple.predict(AreaTumor)

# array([[300.94831572]])

# Tumor pequeño en comparación con los otros grupos de Pacientes

# Si se aumenta el perímetro ([166.82]), el tumor área del tumor se aumenta:

#array([[1939.80435773]])

# Conclusiones

En esta práctica, a través de registros clínicos de cáncer de mama tomados de imágenes digitalizadas de la WDBC (Wisconsin Diagnostic Breast Cancer), se pudo hacer un análisis de estos datos, 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.

* Resultados del algoritmo solo tomando en cuenta las variables seleccionadas

Como se mencionó anteriormente, al aplicar este algoritmo, se obtuvo un **Score de 0.9769**, el cual indica que el **pronóstico del Área del tumor se logrará con un 97.69% de efectividad.**

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

En adición a lo anterior, los pronósticos del modelo final se alejan en promedio **3083.26** y 55.53 unidades del valor real, esto es, **MSE** y RMSE, respectivamente.

* Resultados del algoritmo solo tomando en cuenta todas las variables

Como se mencionó anteriormente, al aplicar este algoritmo, se obtuvo un **Score de 0.9780**, el cual indica que el **pronóstico del Área del tumor se logrará con un 97.8% de efectividad.**

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

En adición a lo anterior, los pronósticos del modelo final se alejan en promedio **2932.75** y 54.15 unidades del valor real, esto es, **MSE** y RMSE, respectivamente.