José Calderón

Universidad Cenfotec

Abstracto

Análisis de dataset para predecir tumores de seno benignos o malignos según información de la paciente

Data Science Capstone Project

Breast Cancer Analysis and Diagnosis

# Introducción

El siguiente Proyecto consiste en buscar un dataset de cualquier tema, plantearnos una pregunta y responderla a través de un modelo de predicción.

En mi caso elegí un dataset de diagnóstico de tumores de senos, esto debido a que es uno de los principales tumores que presentan las mujeres, y en mi familia ya se han presentado varios casos.

# Información del Dataset

El dataset se puede encontrar de manera gratuita en diferentes sitios de internet, y fue creado por el doctor William H. Wolberg de la *University of Wisconsin Hospital, en Madison*. Para este ejercicio se descargó de: <https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>

# Objetivo

Diagnosticar pacientes con cáncer de seno a través del análisis de los datos de los pacientes, y clasificarlos en dos categorías de diagnóstico, Tumor Benigno (B) o Tumor Maligno (M).

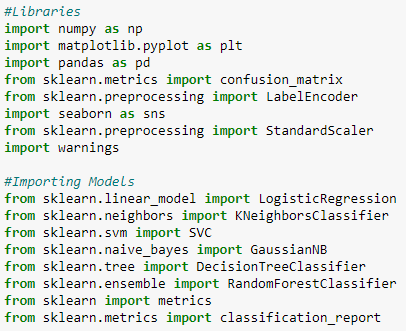
# Colección de Datos

El dataset cuenta con 569 registros y 33 atributos, los cuales son los siguientes:

1. **Id**: Identificador de la fila.
2. **Diagnosis**: Diagnóstico de los tejidos del seno (M = Maligno, B = Benigno).
3. **radius\_mean**: Media de distancias desde el centro a puntos en el perímetro.
4. **texture\_mean**: Desviación estándar de valores de escala de grises.
5. **perimeter\_mean**: Tamaño medio del tumor central.
6. **area\_mean**: Área del tumor central.
7. **smoothness\_mean**: Media de variación local en longitudes de radio.
8. **compactness\_mean**: Media del perímetro ^ 2 / área - 1.0.
9. **concavity\_mean**: Media de gravedad de las porciones cóncavas del contorno.
10. **concave points\_mean**: Media para el número de porciones cóncavas del contorno.
11. **symmetry\_mean**: Media en la simetría del tumor.
12. **fractal\_dimension\_mean**: Media de las dimensiones del fractal.
13. **radius\_se**: Error estándar para la distancia media desde el centro al punto en el perímetro.
14. **texture\_se**: Error en la desviación estándar de la escala gris de valores.
15. **perimeter\_se**: Error estándar en el perímetro.
16. **area\_se**: Error estándar en el área.
17. **smoothness\_se**: Error estándar para la variación local en la longitud del radio.
18. **compactness\_se**: Error estándar para el perímetro ^2 / área - 1.0.
19. **concavity\_se**: Error estándar para la severidad de las porciones cóncavas del contorno.
20. **concave points\_se**: Error estándar para el número de porciones cóncavas del contorno.
21. **symmetry\_se**: Error estándar en la simetría.
22. **fractal\_dimension\_se**: Error estándar para las dimensiones fractales.
23. **radius\_worst**: Peor o mayor valor medio para la media de distancias desde el centro a puntos en el perímetro.
24. **texture\_worst**: Peor o mayor valor medio para la desviación estándar de los valores de escala de grises.
25. **perimeter\_worst**: Peor o mayor valor medio para los perímetros.
26. **area\_worst**: Peor o mayor valor medio para las áreas.
27. **smoothness\_worst**: Peor o mayor valor medio para la variación local en la longitud del radio.
28. **compactness\_worst**: Peor o mayor valor de longitud para perímetro^2 / área - 1.0
29. **concavity\_worst**: Peor o mayor valor medio para la gravedad de las partes cóncavas del contorno.
30. **Concave\_points\_worst**: Peor o mayor valor medio para el número de porciones cóncavas del contorno.
31. **symmetry\_worst**: Peor o mayor valor medio para la simetría.
32. **fractal\_dimension\_worst**: Peor o mayor valor medio para el fractal.
33. **Unnamed: 32**

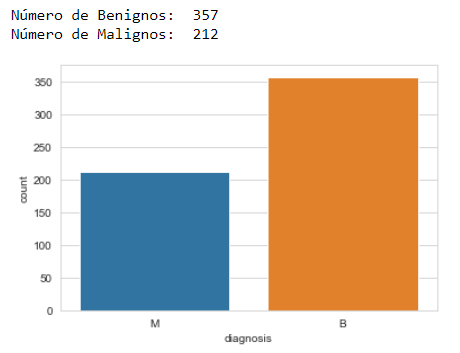
# Librerías Utilizadas

El problema se resolvió utilizando el lenguaje de programación Python, y las librerías utilizadas fueron las siguientes:

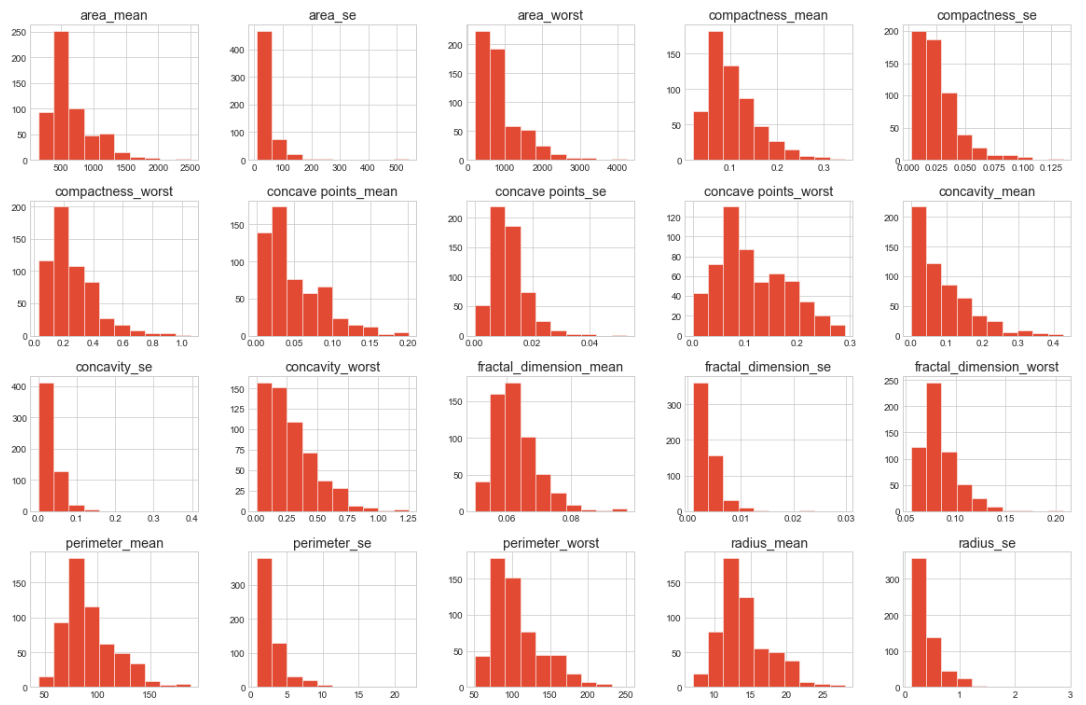


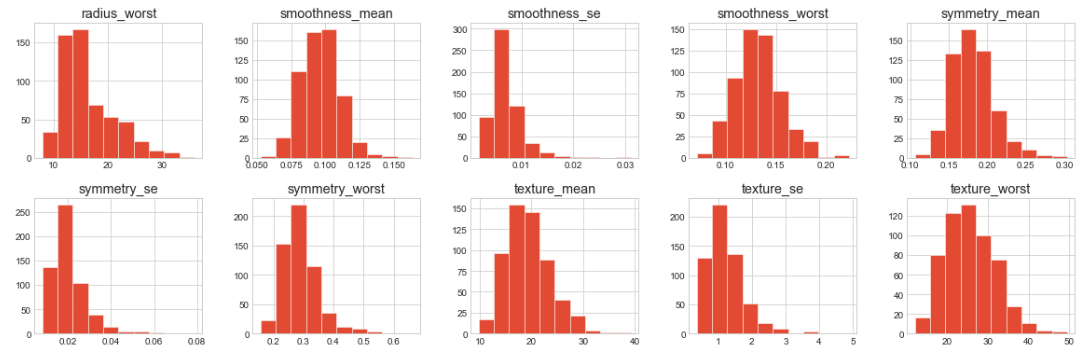
# Pre-procesamiento y Visualización

La distribución de las clases se encontraba de la siguiente manera:

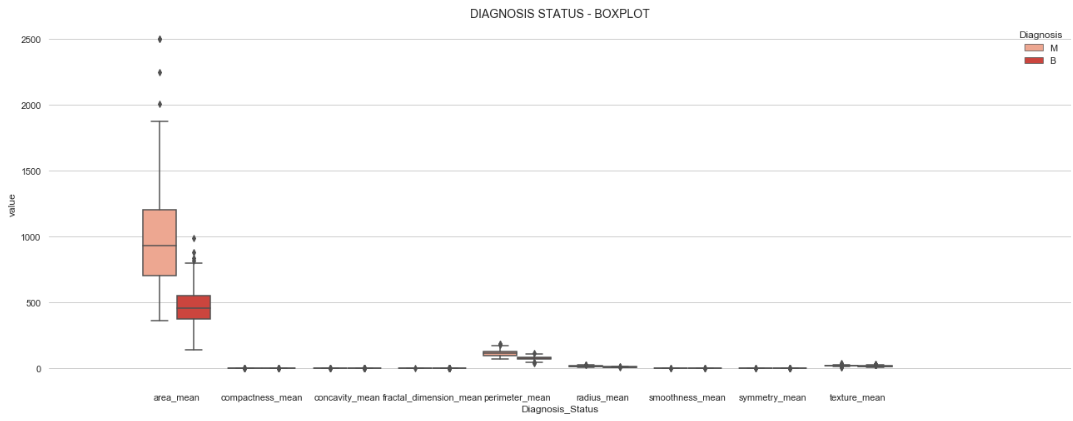


Visualizando la distribución de la data basado en la frecuencia de valores únicos, se obtienen los siguientes gráficos:





Para visualizar la distribución de los valores de los atributos y reconocer los valores atípicos se generó el siguiente gráfico:



Los atributos que se tomaron en cuenta en el gráfico anterior fueron únicamente los promedios y se puede observar que el área es el valor que se encuentra fuera de los rangos que presentan los demás atributos.

# Modificar Atributos

El atributo *diagnosis* fue modificado para pasar de String a valor numérico, ya que es una conversión necesaria para poder trabajar el modelo. La conversión se realiza quedando los valores de la siguiente manera:

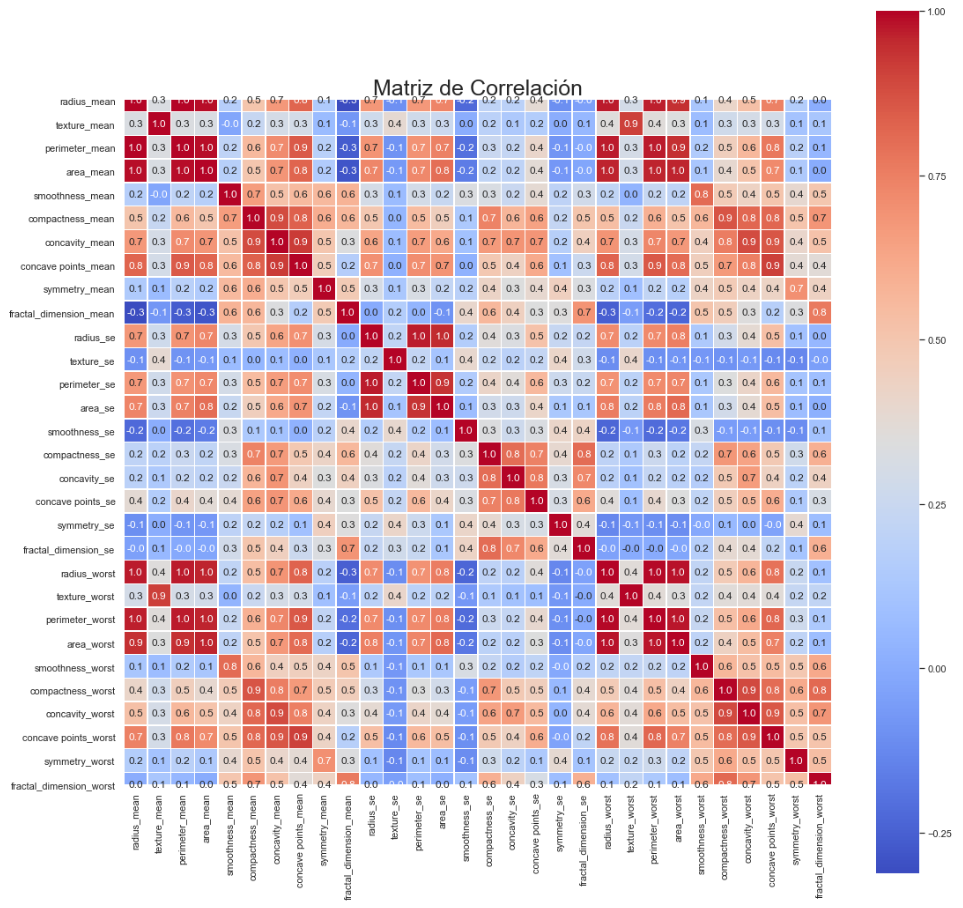
* Maligno (M) = 1
* Benigno (B) = 0

Entre las validaciones realizadas se encontraron valores nulos en el atributo *Unnamed: 32*, por lo que se procedió a eliminar dicho atributo.

El atributo ID también fue eliminado, ya que no aporta valor al momento de realizar el análisis.

# Matriz de Correlación

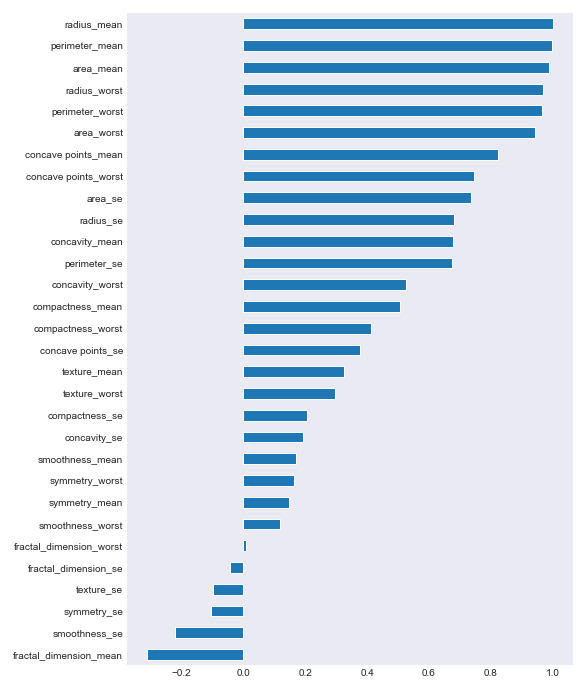
Para determinar la correlación entre los diferentes atributos, se genera el siguiente gráfico:



Se pueden observar que existe una alta correlación (identificada con el valor 1, o el rojo más oscuro) en algunos atributos. Estos valores son:

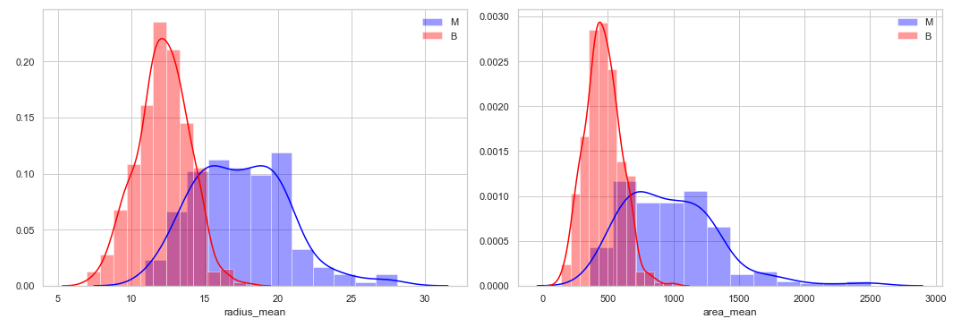
* Radius\_mean
* Perimeter\_mean
* Area\_mean
* Radius\_worst
* Perimeter\_worst
* Area\_worst

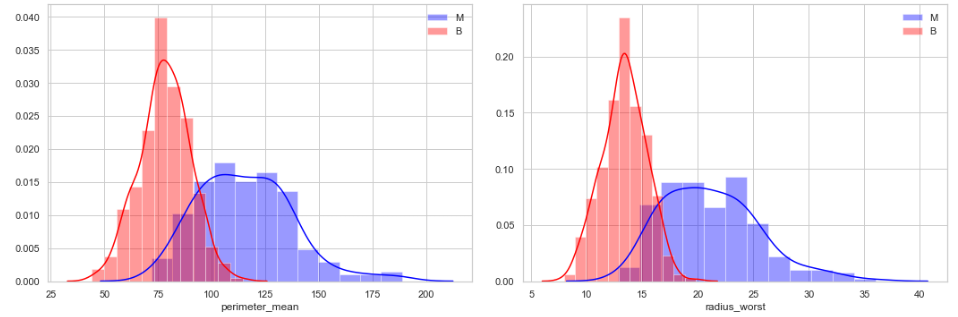
Otra forma de identificar los atributos correlaciones es con el siguiente gráfico:

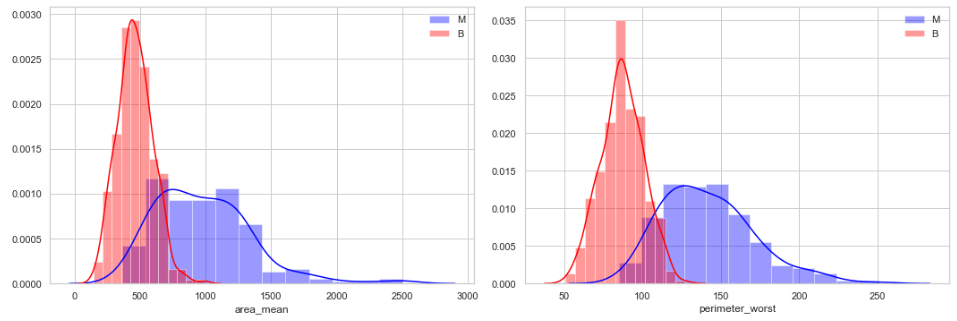


En el gráfico anterior se puede observar con mayor claridad los atributos que presentan una correlación directa.

Los gráficos siguientes muestran la relación existente entre la variable independiente (diagnosis) y los atributos con mayor correlación.







# Procesamiento de Modelos

Para la solución del problema se ejecutaron los siguientes modelos de predicción:

* Logistic Regression
* KNeighbors Classifier
* SVC Linear
* SVC RBF
* Gaussian NB
* Decision Tree
* Random Forest

A continuación, se muestra una tabla con los resultados obtenidos para un entrenamiento de un 70%, y dejando un dataset de prueba de un 30%

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **Precision** |
| Logistic Regression | 97.66 | 98 |
| KNeighbors Classifier | 95.90 | 96 |
| SVC Linear | 95.90 | 92 |
| SVC RBF | 97.66 | 98 |
| Gaussian Naive Bayes | 91.23 | 86 |
| Decision Tree | 93.57 | 88 |
| Random Forest | 96.49 | 98 |

Al modificar el porcentaje de testeo a un 20%, los valores cambian, quedando la tabla de la siguiente manera:

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **Precision** |
| Logistic Regression | 96.49 | 96 |
| KNeighbors Classifier | 95.61 | 100 |
| SVC Linear | 98.25 | 98 |
| SVC RBF | 98.25 | 100 |
| Gaussian Naive Bayes | 90.35 | 88 |
| Decision Tree | 92.98 | 90 |
| Random Forest | 97.37 | 98 |

En el primer cuadro, con una muestra de un 70%, se obtuvieron 2 modelos con los mismos resultados, Logistic Regression y SVC RBF, sin embargo, cuando se volvió a ejecutar el proceso con una muestra de 80%, el resultado fue más favorable para el modelo SVC RBF, por lo cuál es el modelo elegido para este proyecto.