UNIVERSIDAD DE GRANADA

Inteligencia de Negocio
grado en ingeniería informática

Práctica 2

Visualización y segmentación

Autor
José María Sánchez Guerrero

Rama
Sistemas de Información

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2020-2021

Índice

1. Visualización

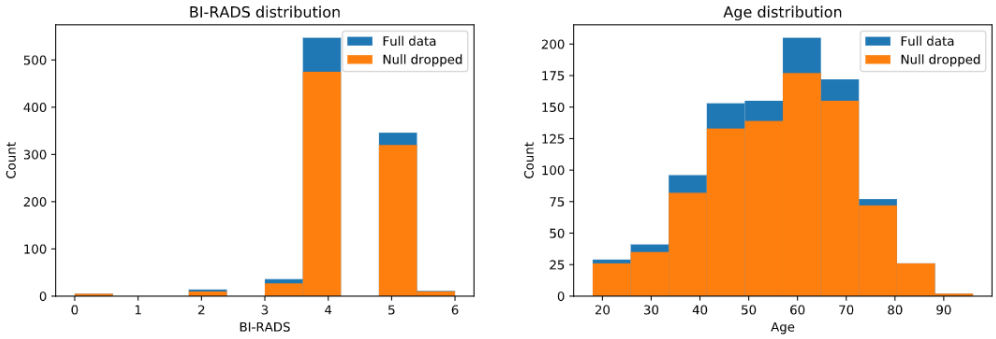
Partiendo del dataset de mamografías de la práctica anterior, tendremos que realizar distintas visualizaciones y analizar los datos que tenemos para cada uno de los preprocesamientos. En mi caso, yo ya mostré en mi práctica anterior una pequeña visualización de los datos (y que los volveré a mostra para analizarlos mejor), pero en esta completaremos más detalladamente la información.

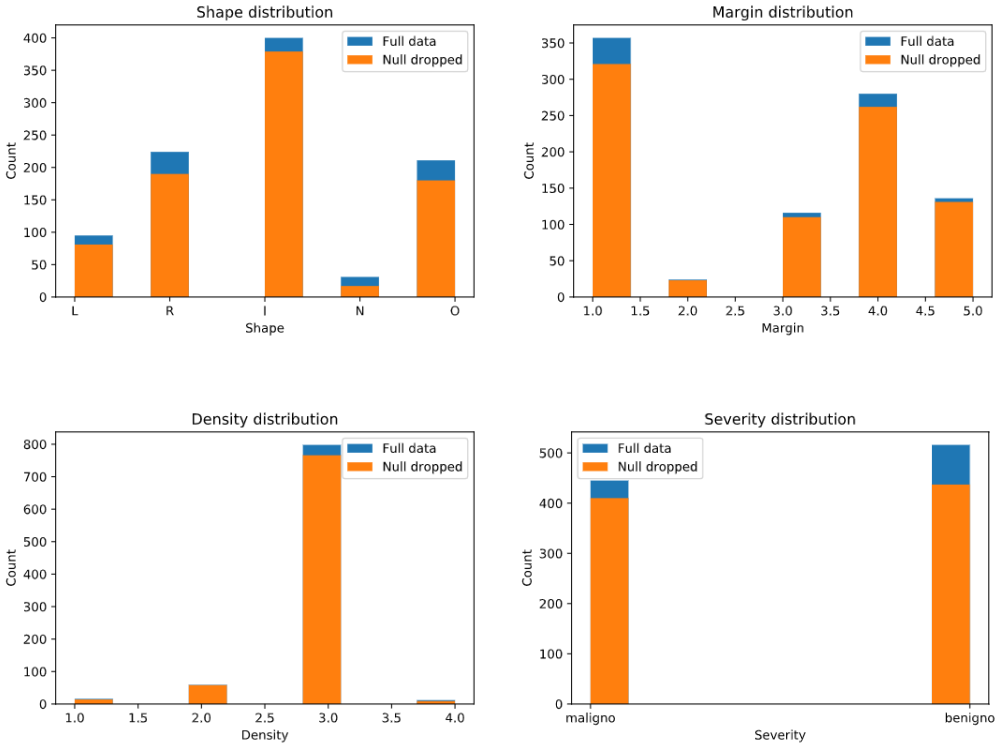
1.1. Visualización de las medidas

Lo primero que pudimos observar, es que tenemos tanto datos numéricos, como datos categóricos. También vimos que hay varias celdas con datos erróneos o perdidos (representados con el valor NaN):

	BI-RADS	Age	Shape	Margin	Density	Severity
0	5.0	67.0	L	5.0	3.0	maligno
1	4.0	43.0	R	1.0	NaN	maligno
2	5.0	58.0	I	5.0	3.0	maligno
3	4.0	28.0	R	1.0	3.0	benigno
4	5.0	74.0	R	5.0	NaN	maligno

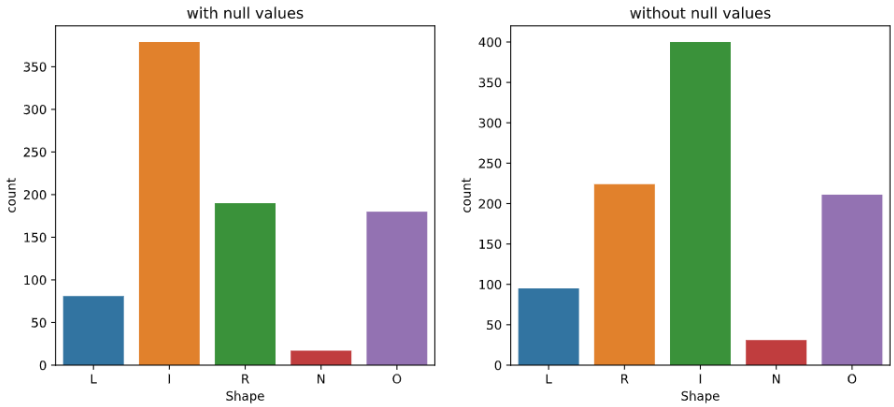
Debido a esto, es importante procesarlos para que nuestros algoritmos funcionen correctamente. La técnica que elegimos en la práctica anterior fue simplemente la de eliminar estos datos erróneos, con el siguiente resultado:





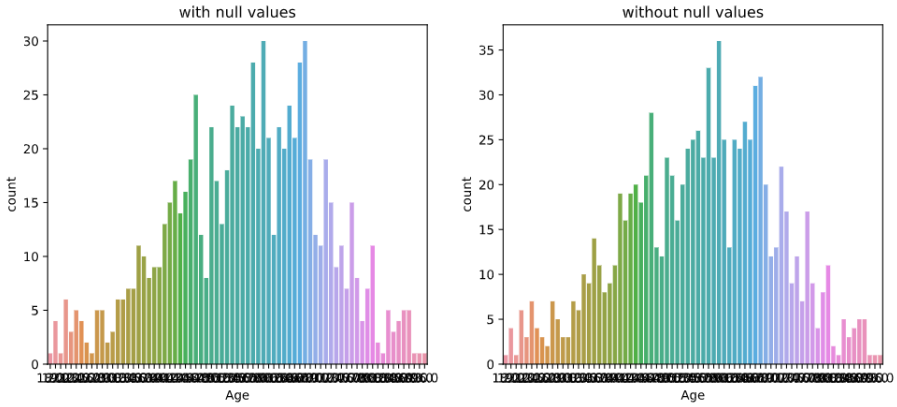
Otra estrategia de preprocesamiento que podríamos haber utilizado es la de completar esos valores con datos. Esto a veces puede no tener sentido, ya que en algunos atributos, como por ejemplo la edad, no vas a asignarle a todos los datos que faltan el valor de 0 años. Para estos casos veremos otra solución más adelante.

Sin embargo, tenemos atributos como la forma ("*Shape*"), que ya de por sí tienen un valor "No definido" con varios datos en él. Gracias a esto, los datos erróneos o nulos del dataset los podemos meter en esta categoría.

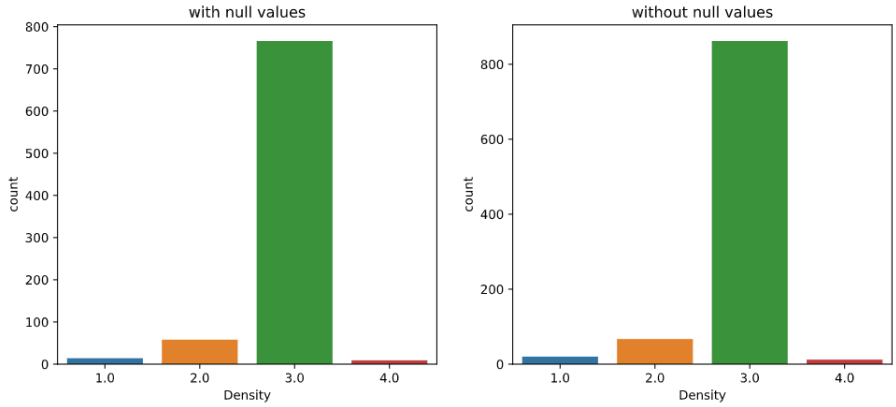


¿Qué estrategia diferente podemos utilizar entonces para el resto sin eliminarlos ni poner valores sin sentido? Tenemos varias opciones. La primera que se nos viene a la cabeza es hacer la media la media entre todos los parámetros y devolver su resultado, pero esto puede *boostear* mucho un valor y darle demasiada importancia. Es una buena medida en caso de que haya pocos datos erróneos o nulos, ya que no va a a desbalancearlo mucho.

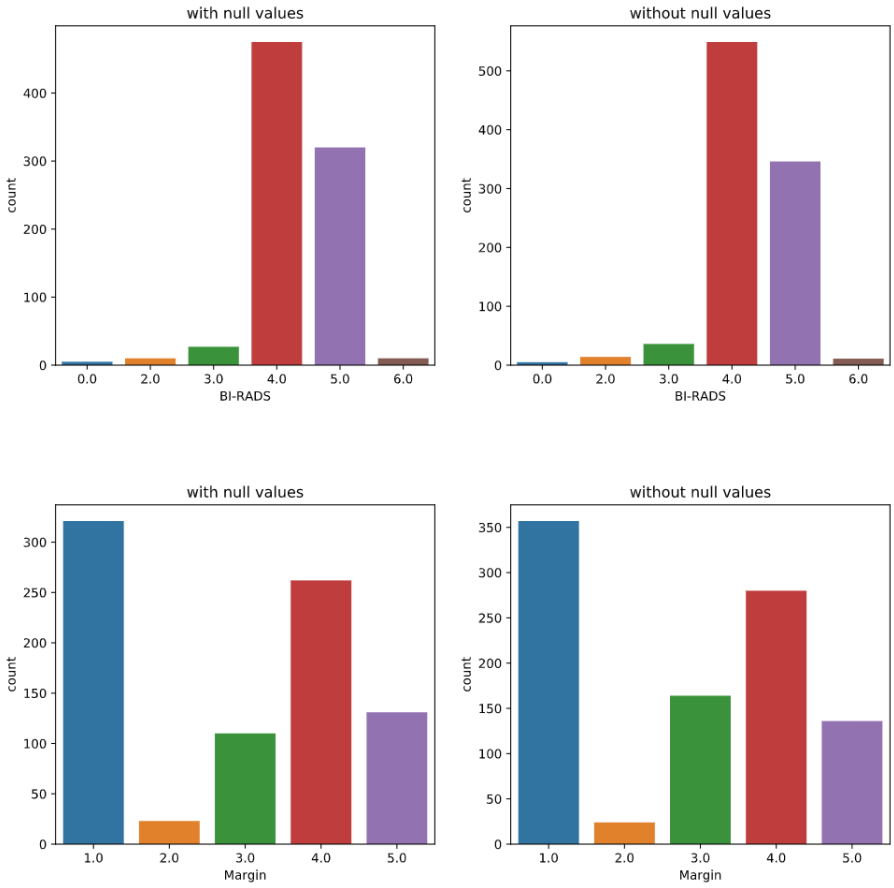
Sin embargo, nosotros vamos a utilizar interpolación con el método '*nearest*' para que tenga en cuenta los valores de los índices más cercanos y así distribuir un poco más los nuevos valores. Los resultados que hemos obtenido para el parámetro edad han sido los siguientes:



Hay que fijarse bien ya que son muchos los valores que puede adoptar la edad, pero vemos cómo no hay ningún valor muy por encima de lo normal ni que hayan sido *boosteados* en exceso. Con otros valores, como la densidad, también lo podemos hacer, pero se aprecia un poco menos la diferencia porque hay menos variedad de datos y el 3 destaca sobre todos los demás.



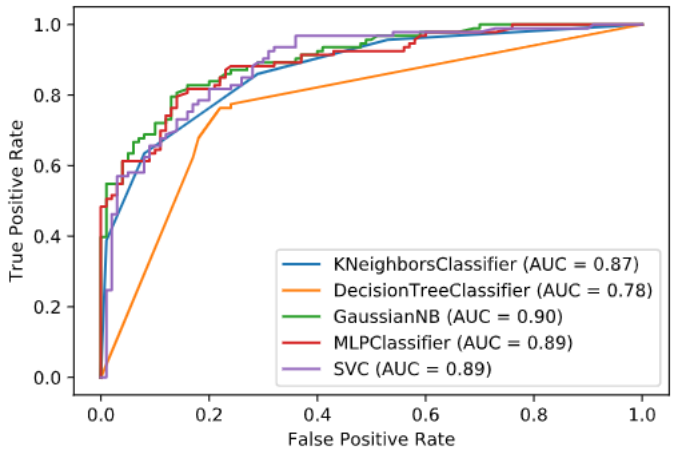
Por último, vamos a hacer una media en los atributos que faltan para poder ver también cómo trabaja esta manera de preprocesado. Podremos observar cómo algunos datos apenas cambian, o es difícil apreciarlo; pero otros, como por ejemplo el valor 3 en 'Margin' sube bastante, llegando al punto de superar el valor 5 y dándole más importancia en el dataset. No quiere decir que esté mal, ya que no se sabe cómo serían esos datos en realidad o si pueden venir más datos como estos en un futuro. Pero hay que tener cuidado porque son datos 'inventados' y, en este caso, originalmente el dataset le da más importancia al 5 que al 3, y lo estamos cambiando.



1.2. Gráficas de curva ROC

No vamos a comentar de nuevo el código, la declaración de los algoritmos, ni la explicación de cómo funcionan, ya que lo hicimos en la práctica anterior. Únicamente decir que se ha utilizado también un '*Label Encoder*', para los atributos que no son numéricos, y que se ha hecho la misma división de los datos.

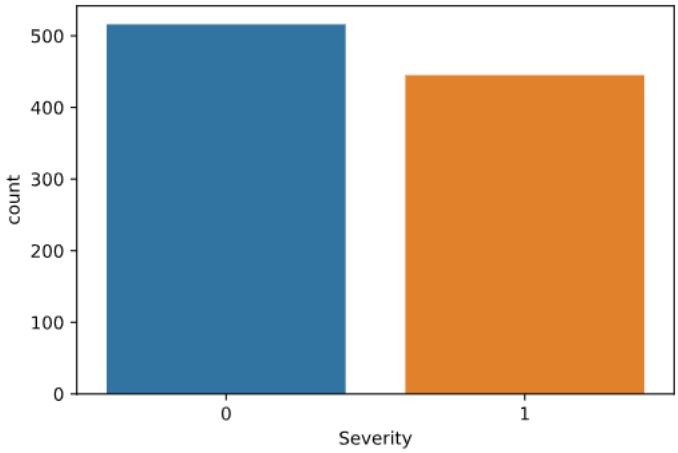
La diferencia con la práctica es que vamos a utilizar simplemente el método *fit()* para entrenar los datos, y después, con el conjunto de test, hacemos todas las curvas ROC y las mostramos en una sóla gráfica. Este ha sido el resultado:



Esta curva representa dos parámetros: tasa de verdaderos positivos y tasa de falsos positivos. El área que hay debajo de la curva (cálculo de la integral) nos dice cómo es el rendimiento del modelo. Es decir, una curva perfecta subiría hasta la coordenada $(0, 1)$, y su integral daría como resultado 1 (100 % de tasa de verdaderos positivos).

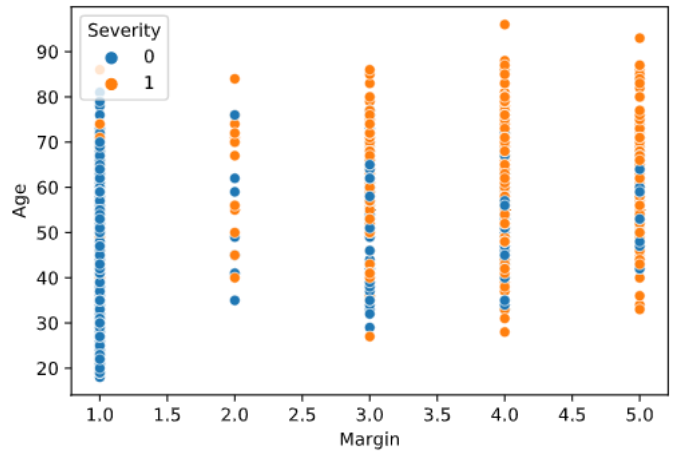
1.3. Análisis de atributos

En esta sección realizaremos una serie de representaciones de los distintos atributos para así intentar encontrar alguna relación entre ellos. Algunos tipos de gráficos ya los hemos utilizado, como puede ser el *countplot*, y el cual lo podemos utilizar simplemente para saber la cantidad de datos que hay, o la cantidad de benignos y malignos que tenemos:



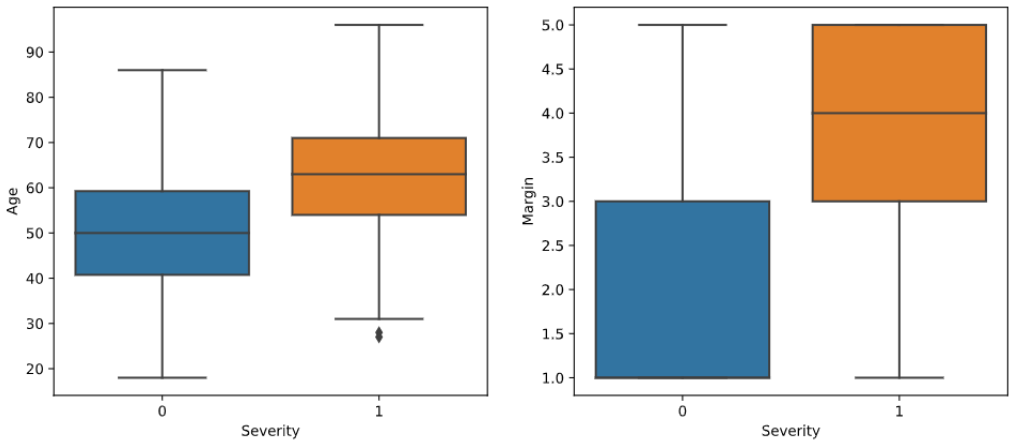
Este gráfico no nos sirve más allá que para contar el número de datos que tiene cada clase, por lo que ya no nos va a servir de mucho tras haber hecho todo lo anterior. Además de que este tipo de gráfico tampoco nos es útil para relacionar 2 o más categorías.

A continuación, vamos a ver uno llamado *scatterplot*, el cual es un diagrama de dispersión que muestra los datos en forma de puntos. La posición de cada uno de los puntos viene determinada por un atributo para cada uno de los ejes. Con el diagrama de dispersión así se podría extraer algún tipo de relación entre los dos atributos, pero para lo que realmente es útil es este tipo de gráficos en *Seaborn*, es por su capacidad de categorizar los puntos con el atributo objetivo, en nuestro caso la severidad. Vamos a ver la relación que existe entre la edad y el margen de masa:



Podemos ver que la relación entre la edad y el margen es prácticamente inexistente, ya que los datos están distribuidos más o menos de forma equilibrada. No obstante, en cuanto a la severidad, si que podemos ver una clara tendencia a ir hacia la izquierda los benignos y a la derecha los malignos. Esto quiere decir que cuanto menor sea el margen de masa, más posibilidades hay de que sea benigno, y viceversa. En cuanto a la edad, lo que vemos no nos dice mucho, ya que hay benignos y malignos repartidos por todas las edades (es cierto que a edades más bajas vemos un poco más de color naranja y a edades más altas azul, pero nada relevante o para poder sacar conclusiones).

En la siguiente gráfica, una *boxplot*, vamos a poder observar mejor mejor la diferencia que hemos comentado anteriormente, porque es un tipo de gráfica (llamada también diagrama de caja) que representa, tanto los cuartiles de los datos con la caja, como el resto de la distribución con los 'bigotes' que salen de ella.



Ahora vemos más claramente lo anterior

Lo bueno es que elimina valores atípicos

2. Procesado de datos

Referencias

[1] Scikit-Learn. *SVC*
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

[2] Scikit-Learn. *RandomForestClassifier*
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier

[3] Scikit-Learn. *MLPClassifier*
https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier

[4] Scikit-Learn. *StandardScaler*
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[5] Scikit-Learn. *PCA*
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[6] Scikit-Learn. *plot_learning_curve*
https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html#sphx-glr-auto-examples-model-selection-plot-learning-curve-p

[7] Scikit-Learn. *recall_score*
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html

[8] Scikit-Learn. *precision_score*
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html

[9] *Should I normalize/standardize/rescale the data?*
http://www.faqs.org/faqs/ai-faq/neural-nets/part2/

[10] Wikipedia. *Sensitivity and specifity*
https://en.wikipedia.org/wiki/Sensitivity_and_specificity

[11] DataFlair. *Kernel Functions-Introduction to SVM Kernel & Examples*
https://data-flair.training/blogs/svm-kernel-functions/

[12] Isaac Changhau. *Loss Functions in Neural Networks*
https://isaacchanghau.github.io/post/loss_functions/

[13] MathWorks*Support Vector Machine*
https://es.mathworks.com/discovery/support-vector-machine.html

[14] Scikit-Learn. *cross_ val_ score*
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

[15] Scikit-Learn. *train_ test_ split*
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[16] Scikit-Learn. *Confusion Matrix*
https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html