Inteligencia de Negocio. Práctica 2: Visualización y Segmentación

Patricia Córdoba Hidalgo
patriciacorhid@correo.ugr.es
Grupo 2 (Viernes)

18 de noviembre de 2020

Índice

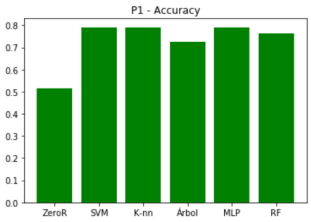
1. Visualización

1.1. Visualización de medidas

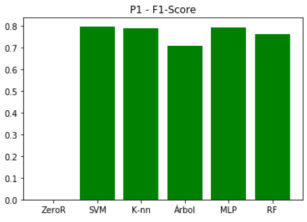
En la práctica 1 se mostraron los datos de cada una de las medidas en tablas, mostrando el valor numérico de éstas. Otra forma de mostrar estos datos es mediante gráficas. En esta práctica, se mostrarán en diagramas de barras los valores de las medidas más utilizadas para la toma de decisiones en la práctica anterior.

Veamos primero los resultados de los diferentes preprocesados de datos:

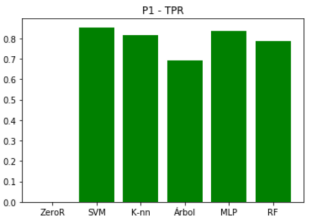
Procesado 1



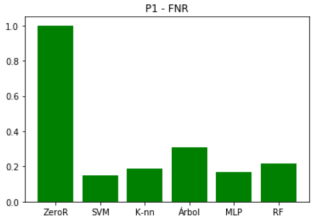
(a) Accuracy



(b) F1-Score

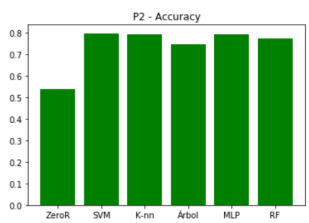


(c) TPR

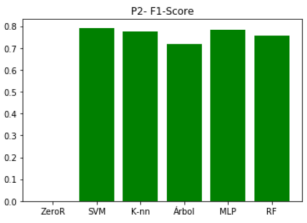


(d) FNR

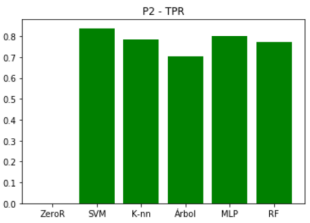
Procesado 2



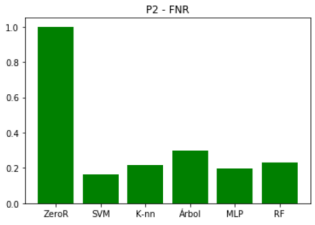
(e) Accuracy



(f) F1-Score

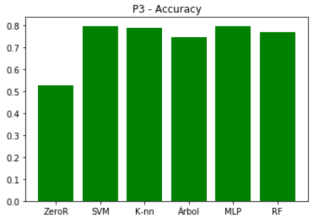


(g) TPR

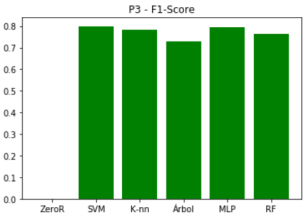


(h) FNR

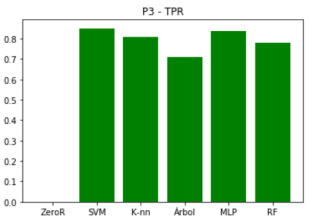
Procesado 3



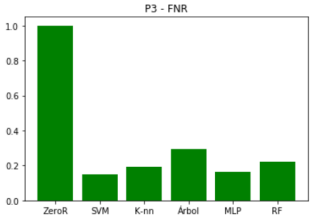
(i) Accuracy



(j) F1-Score



(k) TPR



(l) FNR

El código de cada gráfica es:

```
fig, ax = plt.subplots()
ax.bar(["ZeroR", "SVM", "K-nn", "Arbol", "MLP", "RF"], pX_metrica, color='green')
ax.set_title("PX-metrica")
```

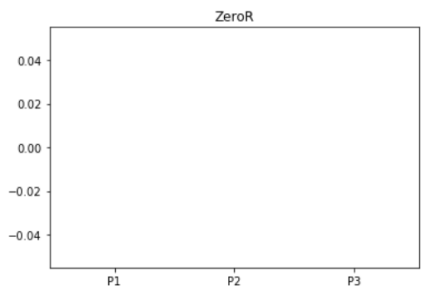
Donde "X" es denota el preprocesado que se ha aplicado, 1, 2 o 3, y " metrica" la métrica que se mide, pudiendo ser `accuracy`, `F1-Score`, `TPR` o `FNR`. El vector pX-metrica es un vector donde se guardan los valores de la métrica "metrica" con el preprocesado "X" de todos los modelos considerados.

Podemos observar que los tres preprocesados obtienen resultados muy similares, como ya comentamos en la práctica anterior. A primera vista, la estructura de los diagramas parece la misma, es decir, al ordenar los diferentes modelos

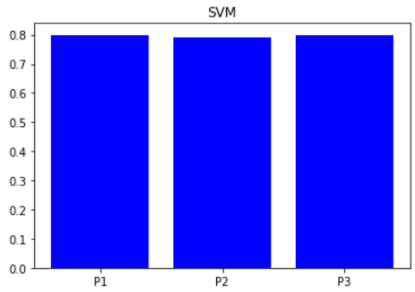
según el valor de la métrica correspondiente con cada procesamiento, este orden es muy parecido en todos ellos, no atreviéndome a decir el mismo por haber barras de alturas semejantes. Es por esto que decantarse por un procesamiento con estos gráficos resulta complicado.

En los tres preprocesamientos el modelo con menor `accuracy` es el ZeroR, seguido del árbol de decisión. El SVM, K-nn y MLP son los modelos con mayor `accuracy` en los tres casos. En el resto de métricas se obserba que el SVM y el MLP tienen un mejor desempeño que el K-nn, siendo estos dos modelos los que mejores resultados obtienen. El árbol de decisión y el ZeroR son los modelos que peor desempeño tienen. El Random Forest y el árbol de decisión no obtienen tan buenos resultados como los otros modelos inicialmente pero, como vimos en la práctica 1, aplicando la poda coste-complejidad obteniamos una gran mejora de su desempeño, ya que con esta poda se conseguía reducir el sobreajuste del modelo.

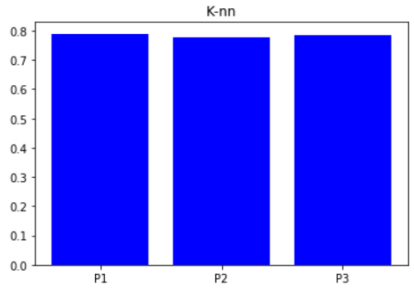
Para elegir que procesamiento utilizabamos en la práctica 1, nos decantamos por el procesamiento que en más modelos tuviese mayor `F1-Score`. En las siguientes gráficas mostramos el valor de esta métrica en los diferentes modelos para cada procesamiento:



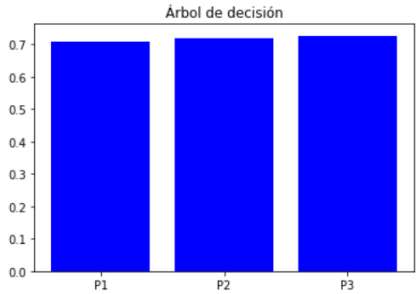
(m) ZeroR



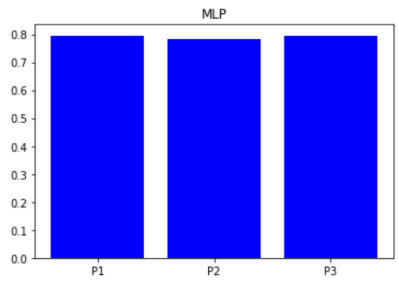
(n) SVM



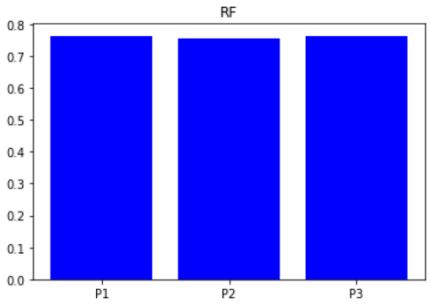
(ñ) k-nn



(o) Árbol de decisión



(p) MLP



(q) Random Forest

Como ya hemos visto antes, no hay gran diferencia entre unos y otros. En el modelo ZeroR la `F1-Score` es 0 en los tres casos, debido a que etiqueta todas las muestras como negaticas, luego no hay verdaderos positivos. En los modelos SVM, MLP, Random Forest y K-nn el procesamiento de datos 2 parece tener resultados ligeramente inferiores que los otros dos, que están muy igualados. En el árbol de decisión el procesamiento 2 tiene un mejor desempeño que el 1, pero peor que el 3. Este modelo es el que peores resultados ofrece sin considerar el ZeroR. Como podemos obsservar que el eje Y no llega a 0.8 como en los demás.

Al igual que en la práctica 1, el procesado de datos que usaría usando la información recogida en estas gráficas sería el procesado 3, porque en el árbol de decisión se ve que es el que mejores resultados ofrece y en el resto de modelos la diferencia con el procesado 1 no puedo apreciarla a simple vista.

Para crear las gráficas primero se crearon para cada modelo vector que contiene el valor de la métrica `F1-Score` de cada uno de los preprocesados:

```
v_zeror = [p1_f1[0], p2_f1[0], p3_f1[0]]
v_svm   = [p1_f1[1], p2_f1[1], p3_f1[1]]
v_knn   = [p1_f1[2], p2_f1[2], p3_f1[2]]
v_arbol = [p1_f1[3], p2_f1[3], p3_f1[3]]
v_mlp   = [p1_f1[4], p2_f1[4], p3_f1[4]]
v_rf    = [p1_f1[5], p2_f1[5], p3_f1[5]]
```

Tras esto, para cada uno de los modelos creamos la gráfica correspondiente así:

```
fig, ax = plt.subplots()
ax.bar(["P1", "P2", "P3"], v_clf, color='blue')
ax.set_title("clf")
```

donde "clf" denota el clasificador del que recogemos los datos en la gráfica.

1.2. Curva ROC

Para representar la curva ROC dividimos el conjunto de datos al que se le ha aplicado el procesado 3 salvo la normalización en comjunto de entrenamiento y conjunto de validación. La división se hace de manera que el 70 % de los datos formen el conjunto de entrenamiento y el otro 30 %, el de test, conservando la proporción de elementos en cada clase tanto en el conjunto de entrenamiento como en el de validación. Esto se hace con el código:

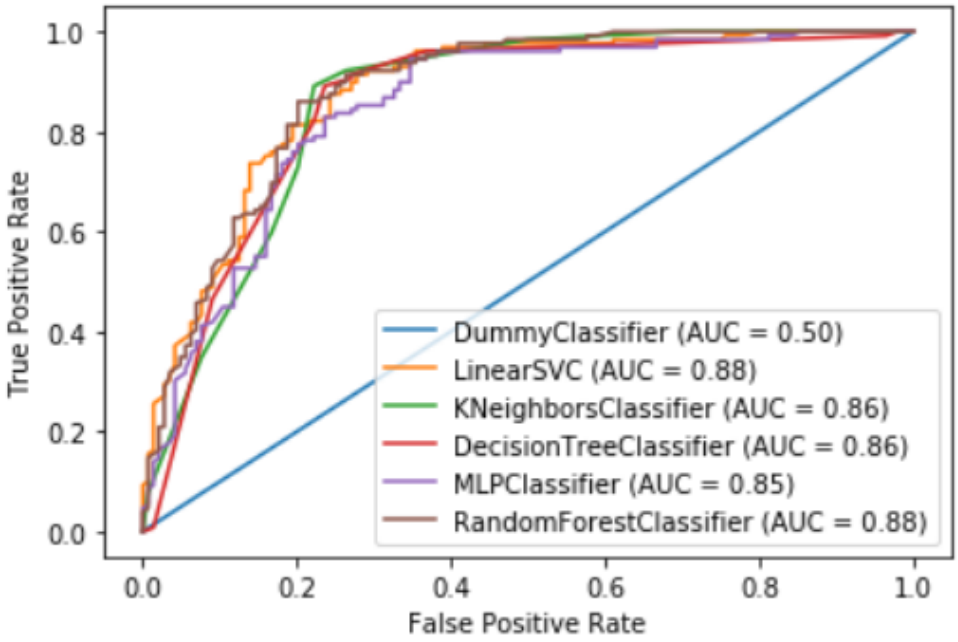
```
X_train, X_test, y_train, y_test = model_selection.train_test_split(data, target,
test_size=0.3, stratify=target, random_state=0)
```

Tras esto,, completamos el procesado de datos 3 usando `MinMaxScaler()` para normalizar los datos de entrenamiento y se usa estas mismas transformaciones sobre los datos de validación. A continuación, entrené los modelos con los hiperparámetros seleccionados en la práctica anterior. Podemos representar en una gráfica la curva ROC de los diferentes modelos así:

```
ax = plt.gca()

for model in [dummy_clf, svm_clf, knn_clf, tree_clf, mlp_clf, rf_clf]:
    metrics.plot_roc_curve(model, data, target, ax=ax)
```

El resultado es:



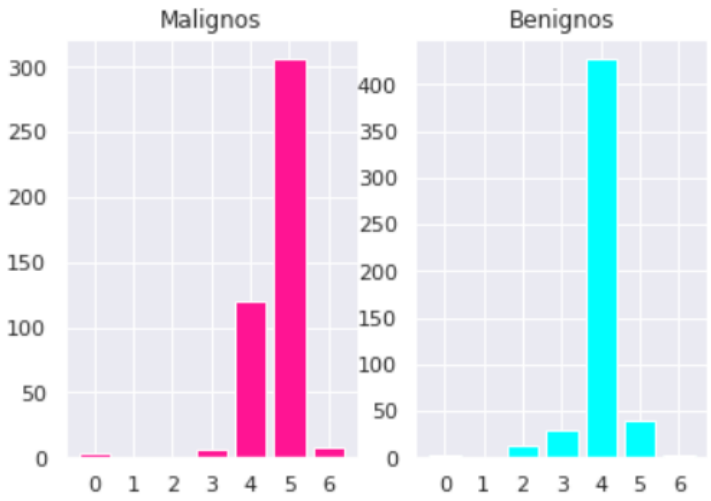
Curva ROC

Los modelos que presentan mayor métrica `AUC` son el Random Forest y el SVM, que son aquellos con mayor área bajo la curva ROC y los que tienen una mayor pendiente en los valores cercanos al cero. Esto implica que estos clasificadores son capaces de incrementar el número de verdaderos positivos a un ritmo mayor que el número de falsos positivos. Si usásemos esta métrica para sacar conclusiones, éstos serían los mejores modelos, mientras que el MLP es el modelo que peor comportamiento muestra, excluyendo al ZeroR. A pesar de esto, no hay excesivas diferencias entre los modelos considerados, a excepción del ZeroR.

1.3. Análisis de los atributos

En esta sección estudiaremos la importancia de los diferentes atributos en la clasificación. Para ello se visualizarán gráficos de barras para cada uno de los diferentes atributos y para cada una de las etiquetas, de manera que se muestre la distribución de los valores que toma dicho atributo en función de su etiqueta. También visualizaremos los diagramas de cajas, "boxplots", de aquellos atributos donde tenga sentido.

Empezamos mostrando las gráficas correspondientes al atributo `BI-RADS`:



Cantidad de datos con cada etiqueta

Este atributo representa la opinión de un médico experto sobre la gravedad del tumor. Si tiene el valor "1" o "2", el tumor es benigno, del mismo modo, si toma el valor "6" es maligno. Los valores "3", "4" y "5" designan casos en los que no se está seguro de la severidad del tumor, pero hay cierta probabilidad de que sea maligno o benigno. Esta información la podemos comprobar en https://es.wikipedia.org/wiki/BI-RADS. Este atributo da mucha información sobre la naturaleza del tumor, pero dado que necesitamos la opinión de un experto para obtenerlo, no es apropiado usarlo para el aprendizaje.

Según la página https://bigml.com/user/TotyB/gallery/dataset/509a98c6035d0706dd0001dd, de donde hemos obtenido los datos, el 94.46 % de estos tienen `BI-RADS` "4" o "5". De estos, los tumores con valor "5" son probablemente malignos. Los tumores malignos tienen más variabilidad que los benignos, dado que hay tumores malignos con valor de `BI-RADS` "4" o "5", predominando el valor "5". En el caso de los benignos, la mayoría de estos tienen la categoría de `BI-RADS` "4".

Esto se ve en el diagrama de cajas de este atributo:

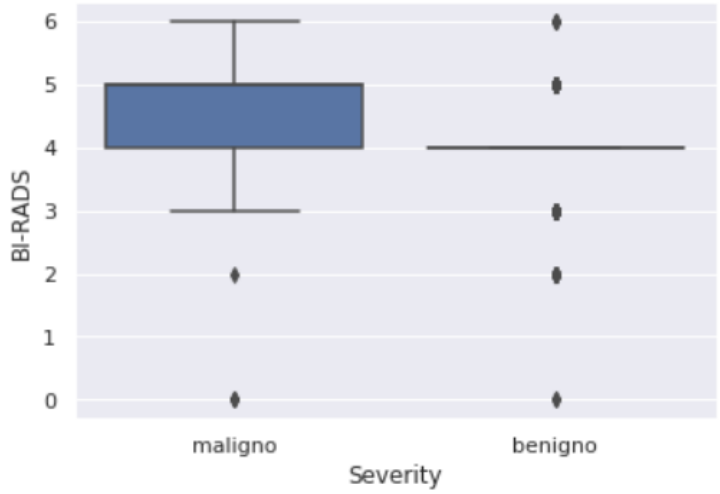
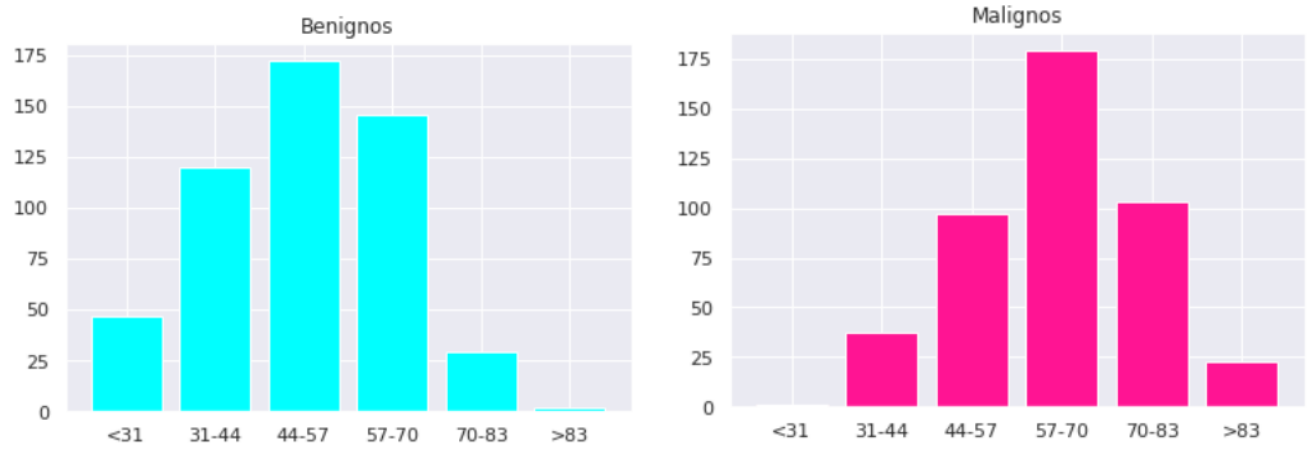


Diagrama de cajas de BI-RADS

Los tumores etiquetados como malignos se mueven entre el valor “4” o “5” mientras que la mayoría de los benignos toman el valor “4”.

La gráfica del atributo edad nos proporciona los siguientes resultados:



Cantidad de datos con cada etiqueta

Vemos que cuanto menor edad tiene una persona, más probable es que su tumor sea benigno. En particular, la mayoría de las personas menores de 31 años de la muestra tienen tumores benignos y gran parte de las que tienen más de 83 tienen tumores malignos. En el diagrama de cajas podemos observar que la mediana de edad de los pacientes con tumores malignos de la muestra es superior a los 60 años, mientras que la de los pacientes con tumores benignos está rondando los 50. Aunque la edad de los pacientes con tumores benignos varía entre los 18 y algo más de los 80 años, los datos entre el primer y el tercer cuartil se encuentran concentrados entre los 40 y 60. Los datos entre el primer y el tercer cuartil de los tumores malignos se encuentran entre algo más de los 50 y algo más de los 70 años.

Por consiguiente, la edad de una persona sí es un atributo relevamte en la clasificación, porque aunque no podríamos estimar la severidad del tumor sabiendo sólo la edad de ésta, si una persona es muy joven podríamos esperar que su tumor sea benigno.

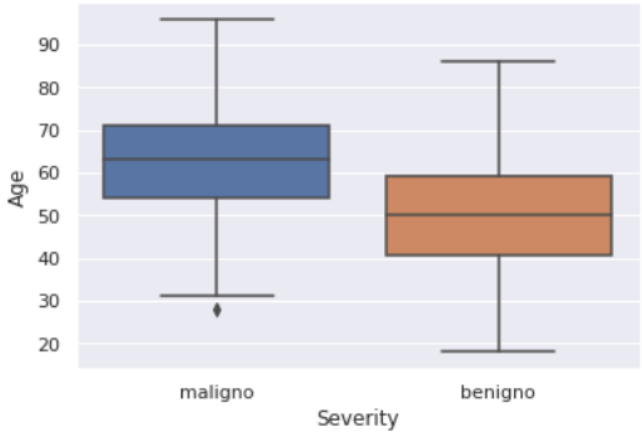
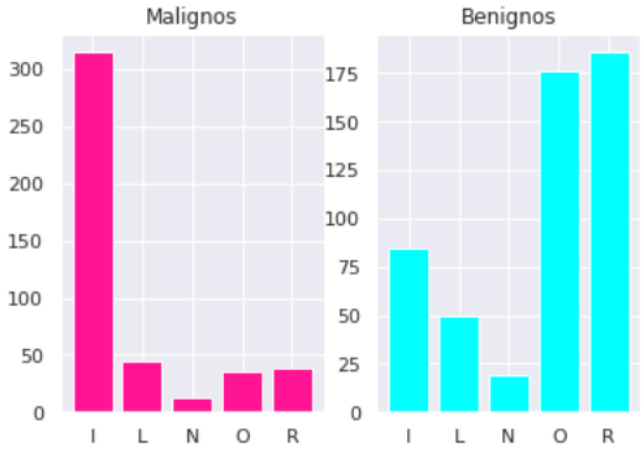


Diagrama de cajas de la edad

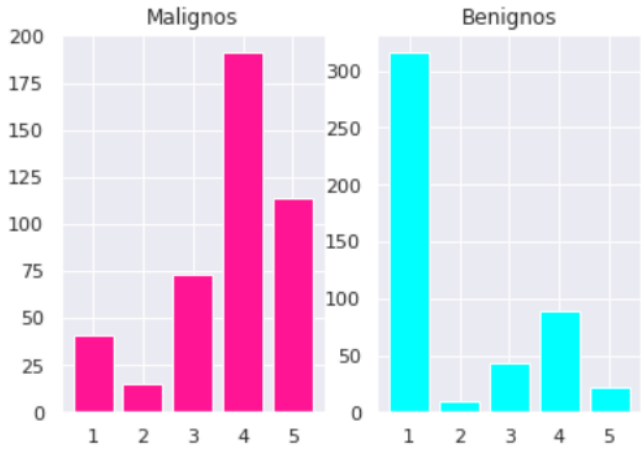
A continuación, analizaremos las gráficas del atributo Shape. La mayoría de los tumores malignos tienen una forma irregular, mientras que los benignos suelen tener una forma ovalada o redondeada. Los tumores benignos presentan mayor variabilidad en los valores que toman en este atributo que los malignos. Resulta interesante considerar este atributo para nuestro aprendizaje, ya que la distribución de los valores que toman los tumores malignos difiere bastante de la de los benignos.



Cantidad de datos con cada etiqueta

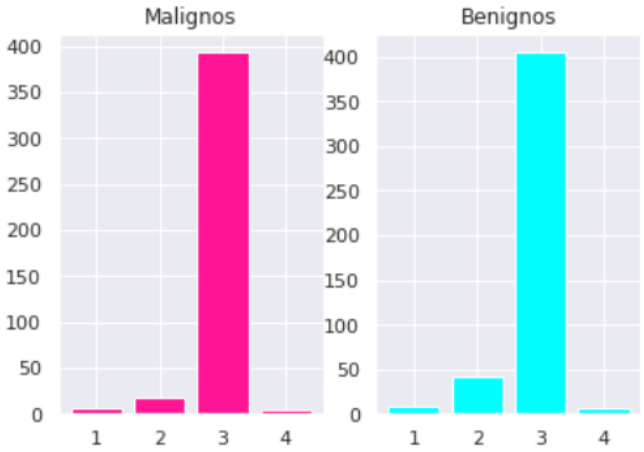
Como el atributo Shape es cualitativo y no tiene orden, no tiene sentido dibujar un diagrama de cajas en este caso, ya que este variaría para cada asignación diferente de valores numéricos a cada posible valor del atributo, y esta asignación es totalmente aleatoria, ya que estos carecen de orden.

El siguiente atributo a analizar es Margin. Al igual que en Shape, los distintos valores que toma este atributo no tienen un orden determinado, por lo tanto no representé el diagrama de cajas de este atributo. Los tumores malignos tienen mayor variabilidad que los benignos, siendo el valor de Margin más frecuente entre estos "ill-defined" seguido del "spiculated". Los benignos, sin embargo, suelen tener margin "circumscribed". En la práctica 1, una vez eliminados los atributos `BI-RADS` y Density, vimos que este atributo era el que mayor información nos aportaba en la clasificación, dado que era el elegido en el nodo raíz del árbol de decisión. Aquí podemos comprobar que efectivamente hay una gran diferencia entre la distribución de los valores de Margin que toman los tumores malignos de los benignos, cosa que afecta favorablemente a la clasificación, permitiéndonos diferenciar una muestra maligna de una benigna con mayor facilidad.



Cantidad de datos con cada etiqueta

Por último, analizamos el atributo Density. Como ya comentamos, este atributo tiene muy poca variabilidad en la muestra y podemos comprobar que los dos diagramas de barras son muy parecidos, lo que nos incita a pensar que este atributo no aporta apenas información a la clasificación.



Cantidad de datos con cada etiqueta

En el diagrama de cajas volvemos a apreciar la poca variabilidad de este atributo, ya que cualquier valor distinto de 3 lo interpreta como outlayer. Esta razón fue la que me llevó a considerar eliminar este atributo durante el procesado de datos 3.

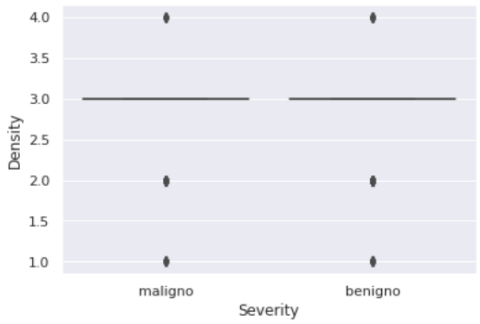


Diagrama de cajas de Density

2. Segmentación