PRÁCTICA MINERÍA DE DATOS: PYTHON

Hernández Sánchez, Víctor

Escuela Politécnica Superior de Elche, Universidad Miguel Hernández

Grado en Ingeniería Informática en Tecnologías de la Información

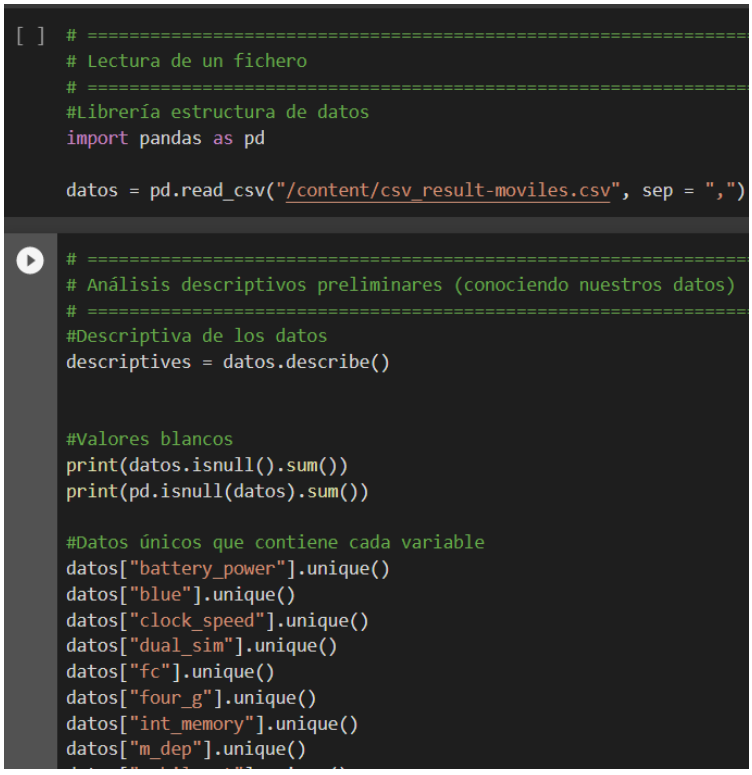
Curso 2022-2023, Minería de Datos

9 de noviembre de 2022

Análisis de Datos con Python

Para comenzar, importaremos la librería panda con la abreviatura pd para mayor comodidad a la hora de implementar este código. Después, leeremos nuestros datos analizados la práctica anterior con Weka, pero esta vez desde un archivo csv en lugar de un .arff.

Seguidamente, describimos todos nuestros datos marcándolos como únicos y escribiendo el nombre de cada columna a usar. Además, imprimimos cuales tienen blancos. En este caso, todos salen a cero, por lo tanto, ninguno está en blanco.

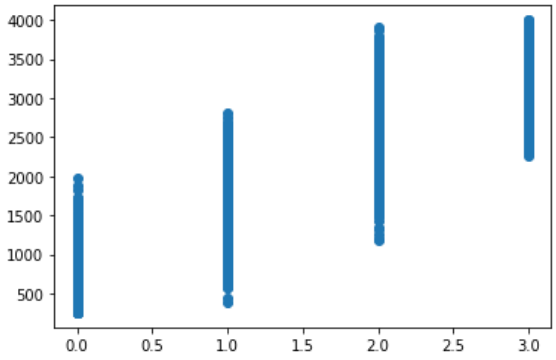


```
[ ]  # ============================================================
     # Lectura de un fichero
     # ============================================================
     #Librería estructura de datos
     import pandas as pd

     datos = pd.read_csv("/content/csv_result-moviles.csv", sep = ",")
```

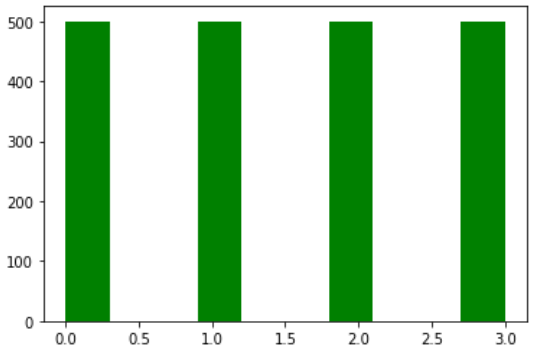
```
     # ============================================================
     # Análisis descriptivos preliminares (conociendo nuestros datos)
     # ============================================================
     #Descriptiva de los datos
     descriptives = datos.describe()


     #Valores blancos
     print(datos.isnull().sum())
     print(pd.isnull(datos).sum())

     #Datos únicos que contiene cada variable
     datos["battery_power"].unique()
     datos["blue"].unique()
     datos["clock_speed"].unique()
     datos["dual_sim"].unique()
     datos["fc"].unique()
     datos["four_g"].unique()
     datos["int_memory"].unique()
     datos["m_dep"].unique()
```

Una vez esto está completado, podemos comenzar nuestro análisis descriptivo mediante gráficos y correlaciones. Para ello, primeramente, importaremos la librería matplotlib, que nos ayudará con los gráficos.

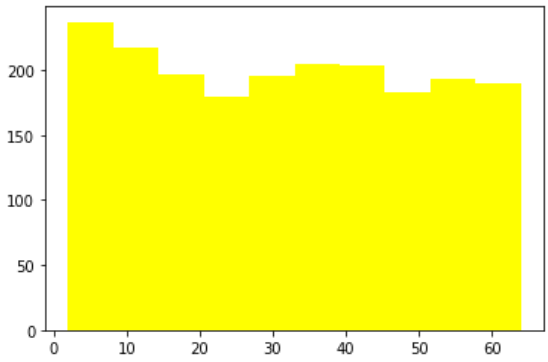
Realizaremos 3 tipos de gráficos distintos, histogramas, gráficos de líneas y de dispersión. En mi caso, se realiza un solo gráfico de dispersión ya que, al tener los datos tan discretizados, no aporta gran información. También tendemos dos gráficos de líneas y tres histogramas.



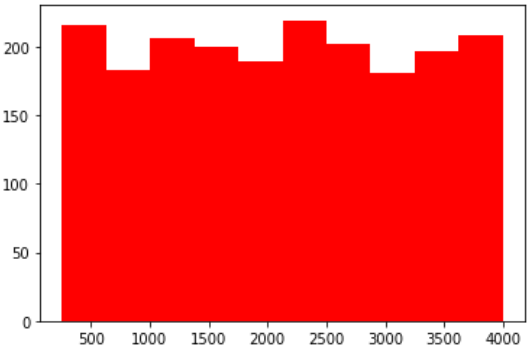
1.Gráfico de dispersión (ram y Price_range).



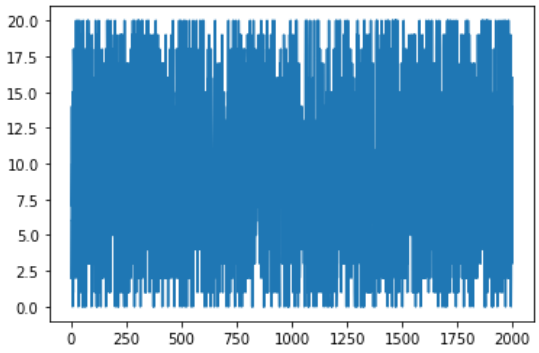
2.Histograma (Price_range).



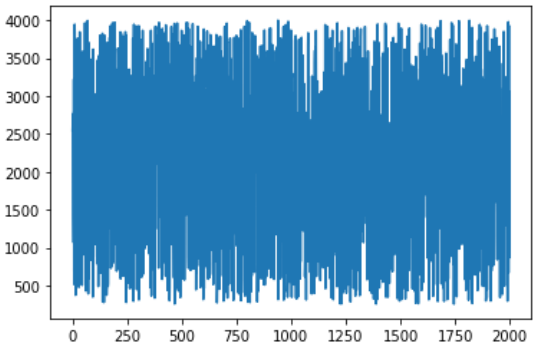
3.Histograma (int_memory).



4.Histograma (ram).



5.Gráfico de barras (pc).



6.Gráfico de barras (ram).

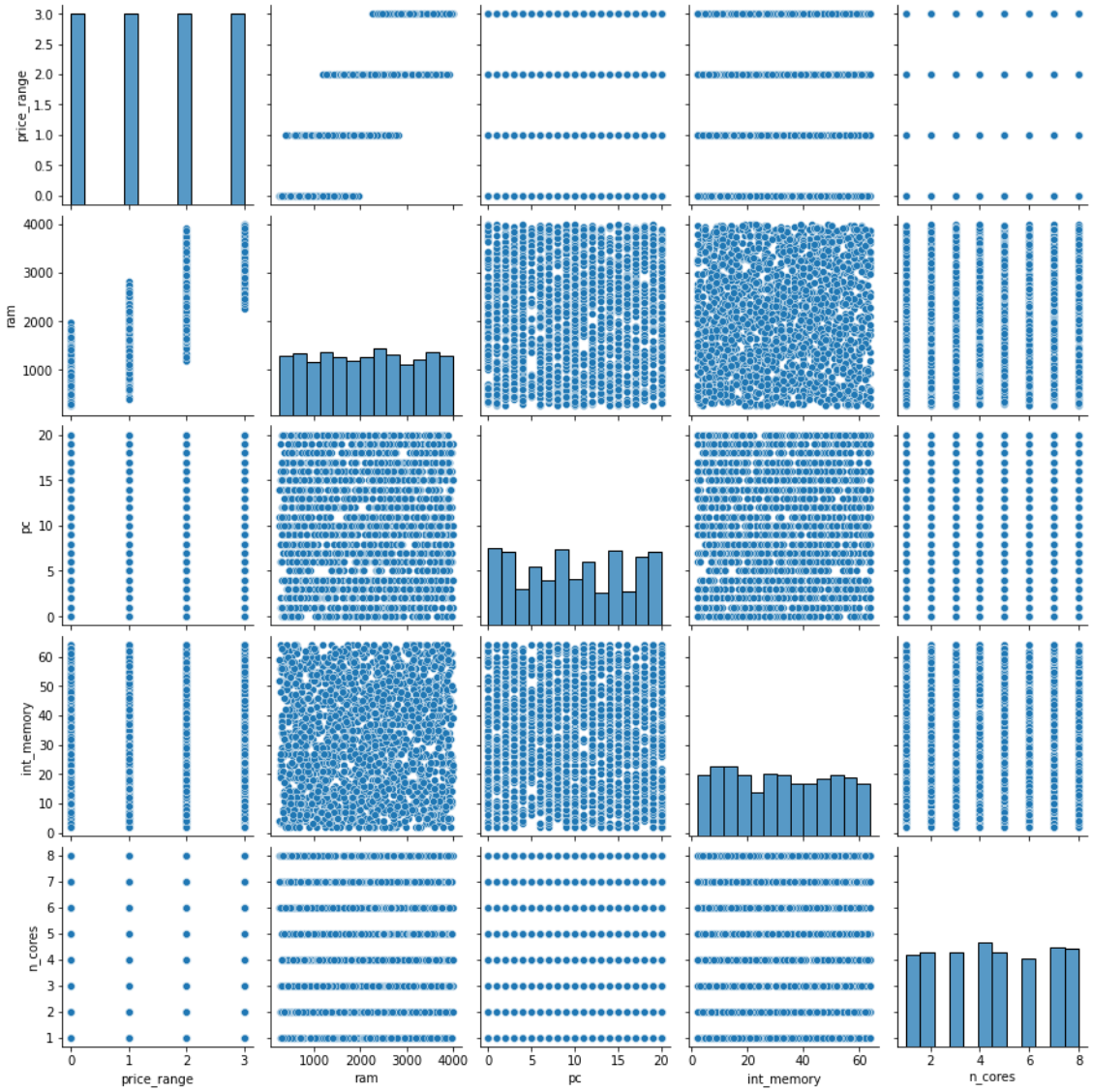
Un análisis de estos gráficos nos indica que cuanto mayor es el rango de precio, mayor será la ram del teléfono (1).

También sabemos que el rango de precios está igualmente repartido entre todas las instancias a estudiar, 500 cada rango (2) y que hay más móviles con una memoria interna baja, pero en general, ninguna destaca en exceso (3).

Así mismo, lo más común es encontrar una ram baja o una de mitad de tabla y que la ram máxima es más común que algunos rangos intermedios, a pesar de que nuevamente todo está bien repartido (4).

Además, en todas las instancias vemos como, en general, encontramos un rango de pixeles de la cámara bastante repartido, lo que indica que podremos encontrar casi cualquier tipo de cámara en los 4 rangos de precio (5) y algo parecido sucede con la ram, aunque se aprecian menos instancias de móviles que, en general, tengan los máximos rangos de ram (6).

A continuación, usaremos también otra librería, llamada seaborn, que nos generará cruces de gráficos de dispersión de todas las variables o del rango que elijamos. En este caso, hemos seleccionado Price_range, ram, pc, int_memory y n_cores.

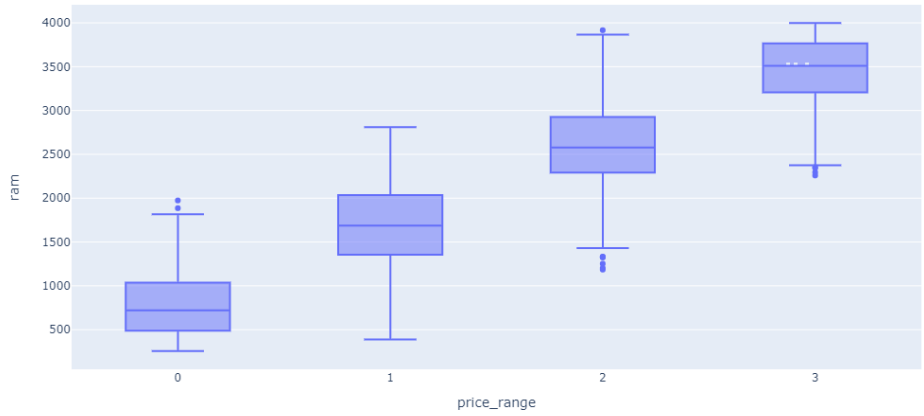


7.Cruce de gráficos de dispersión.

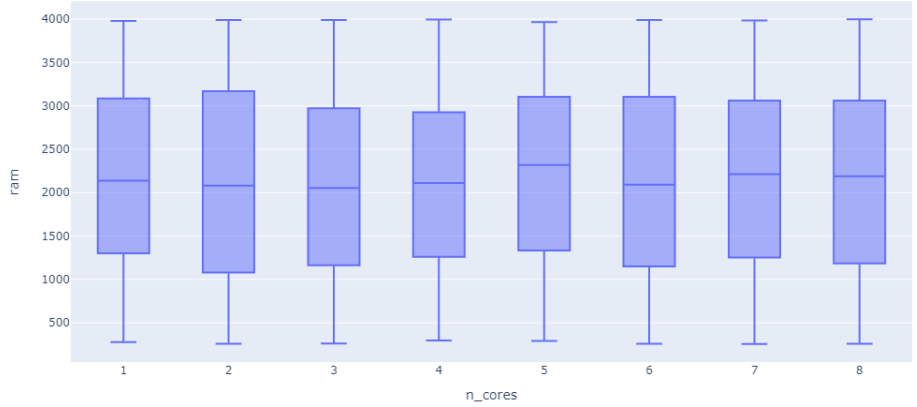
Como vemos, se genera un gráfico en el que filas y columnas son las variables seleccionadas y el cruce de dos da de resultado un gráfico. En el caso de que sean ambas iguales, muestra un histograma de sus instancias (7).

Si lo analizamos, podemos extraer conclusiones como, por ejemplo, que el precio solo influye en la ram, ya que en los demás, podemos encontrar instancias de todos ellos en todos los rangos de precios. También podemos apreciar que los gráficos de n_cores están muy discretizados, ya que sus rangos entre las instancias son muy concretos (del 1 al 8 y números enteros).

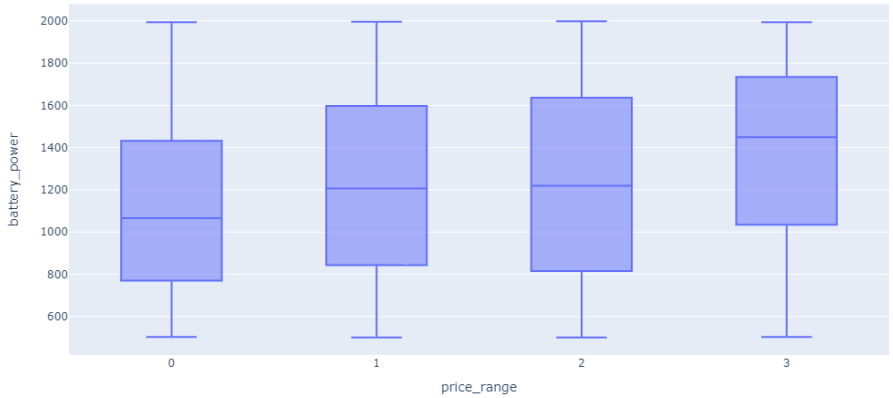
Ahora, añadiremos un tipo de gráfico nuevo usando plotly express. Este gráfico tendrá barras de errores relativos y rangos.



8.Gráfico con barra de errores (ram y Price_range).



9.Gráfico con barra de errores (ram y n_cores).



10.Gráfico con barra de errores (battery_power y Price_range).

Con estos gráficos, podemos ver resultados parecidos a los obtenidos anteriormente, pero, por ejemplo, vemos que en cuanto a ram por rango de precio se refiere (8), los rangos son mucho más marcados ahora.

También, en los otros dos gráficos (9) (10), vemos que el rango de error va desde arriba hasta abajo, lo que nos dice que los datos son bastante difusos y no se tiene una conclusión muy concreta, pero podemos apreciar, por ejemplo, que a mayor ram, mayor rango de cores tendremos o que cuanto más rango de precio, más batería en general tendrá nuestro móvil.

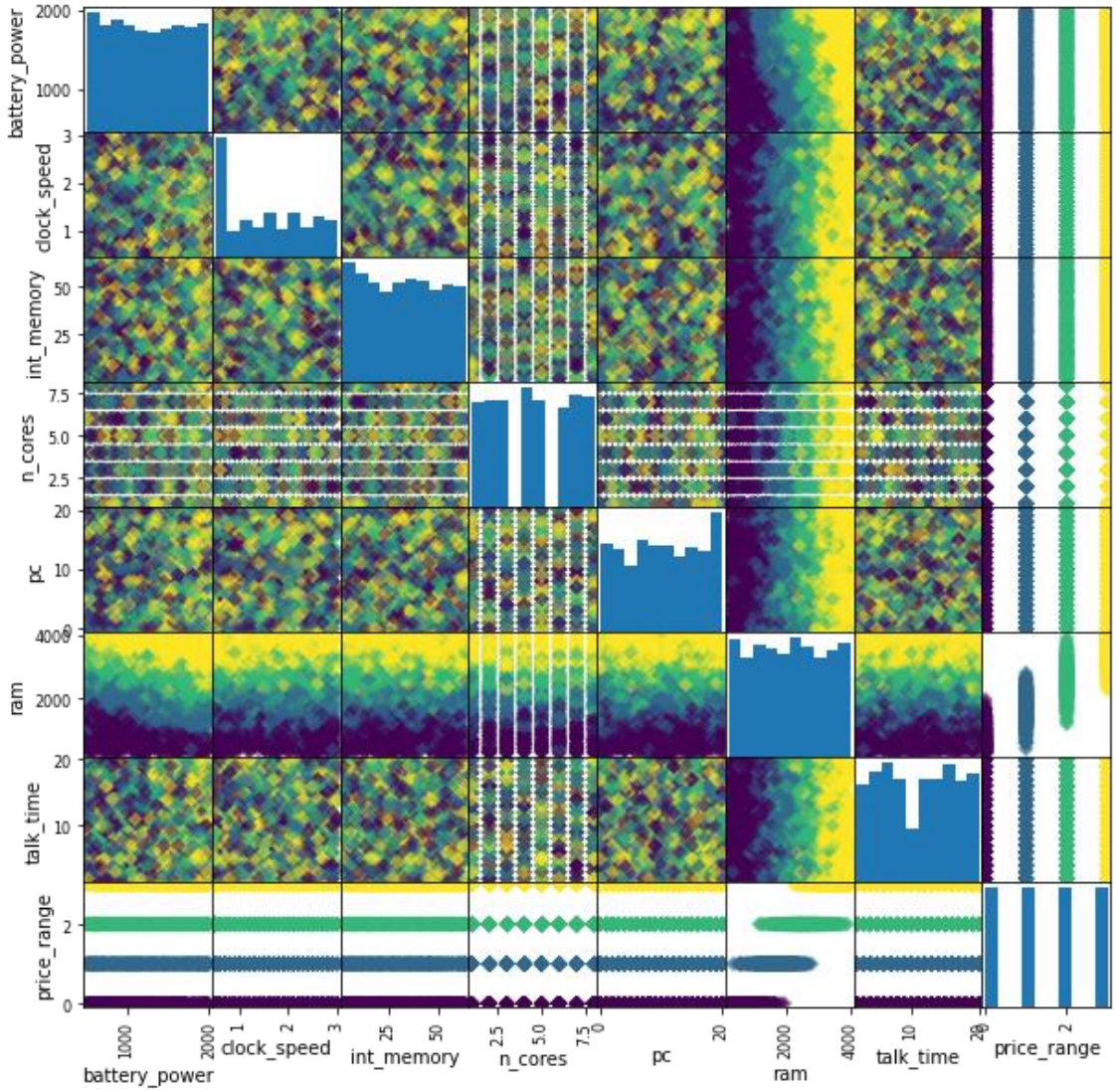
Después de esto, tenemos la librería de ploltly offline, la cual como resultado nos generará un documento html con nuestro gráfico. Gracias a esto, podremos generar gráficos a nuestro gusto introduciendo los datos que queramos trazar.

Ahora, trataremos los datos para mejorar los resultados del siguiente gráfico.

Descartaremos algunas de las variables menos importantes y quitaremos los registros que sean nulos si los hubiera.

También convertiremos nuestra variable objetivo a numérica, ya que esta es nominal.

Finalmente, tendremos el gráfico, una matriz de dispersión similar a la creada anteriormente, solo que en este caso le indicaremos una serie de parámetros distintos como el color, el marcador y la forma de marcar o el tamaño de la ventana.



11. Matriz de dispersión.

Si analizamos los datos de esta matriz (11), podremos extraer las mismas conclusiones que anteriormente, solo que se aprecia mucho más sencillamente gracias a los 4 colores de los 4 rangos del precio. Además, hay alguna variable más incluida, como, por ejemplo, clock_speed o talk_time.

Para continuar, seguiremos con métodos supervisados. En este caso, con árboles de decisión mediante las librerías de sklearn y matplotlib y haremos una clasificación en un árbol.

El índice de este árbol será: battery_power, int_memory, n_cores, ram, talk_time y Price_range (variable objetivo). La imagen se incluye con el siguiente vínculo para que pueda apreciarse correctamente.

https://drive.google.com/file/d/1hruZRJjZ_Y3Yeww9pFCZ_FnLgs30IZ06/view?usp=sharing

Ahora, usando el mismo método, haremos un árbol que nos servirá de predicción en lugar de clasificación. Lo entrenaremos primeramente y después lo imprimiremos ya entrenado.

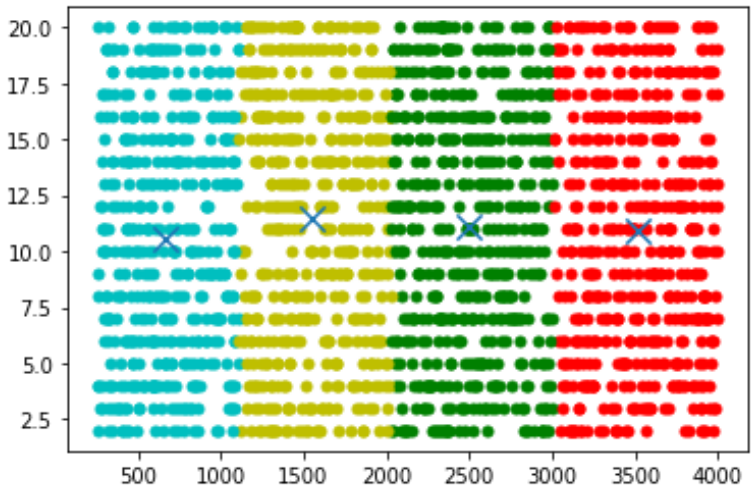
https://drive.google.com/file/d/1NsQneIjqi0HlALEJ7dE7tYyni86pU9TI/view?usp=sharing

Para finalizar con los árboles, también usaremos el método de regresión, que también usaremos los datasets de entrenamiento y de prueba para lograrlo.

https://drive.google.com/file/d/1MJsDLUkXshNmSJjYJvLqtYEpoYJ60j5/view?usp=sharing

Ahora, usaremos métodos no supervisados para la siguiente parte de la práctica. En este caso, usaremos el Clustering también con la librería sklearn. Para este gráfico, seleccionaremos las variables ram, talk_time, battery_power y Price_range.

Generearemos los centroides de cada cluster y pintaremos de cuatro colores distintos para diferenciarlos (12).

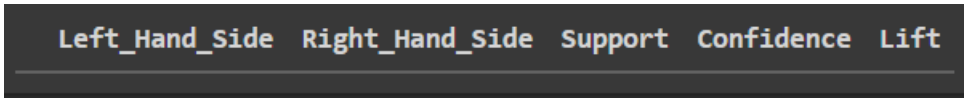


12. Clustering con kmeans (4 clusters).

Para finalizar la práctica, aplicaremos reglas de asociación con A priori usando numpy, pandas y apyori.

Los datos que seleccionaremos serán blue, dual_sim, fc y Price_range. Lo imprimiremos todo con pandas y tendremos los parámetros: left_hand_size, right_hand_size, support, confidence y lift.

El problema que nos surge es que, con este Dataset, los métodos de asociación no funcionan correctamente, por tanto, no se logra ninguna regla lo suficientemente buena como para mostrarse (13).



13.Parámetros de predicción de A priori.

Podemos sacar, como conclusión, que el análisis de los datos es muy parecido al que extraemos con Weka, solo que, haciéndolo de esta forma, se imprime de una forma más bonita, cómoda y sencilla de visualizar, analizar y entender.

Además, todos los gráficos y colores en ellos nos ayudan a entender los datos mucho más fácilmente.