

Reporte - Actividad 7

García Parra Joel Alberto

Marzo 2019

Esta actividad se centra más en aprender un poco de una nueva librería, la cual es *seaborn*, en donde se aprendió de manera sencilla a la elaboración de un *heatmap* en base a la comparación de una serie de datos, donde el problema realmente es la comparación de dichos datos.

1. Comparación de los datos

Aquí se utilizó a la librería *re* para poder encontrar denotar las columnas que se tenían que eliminar, de la siguiente forma:

```
var = []
for m in range(0, len(df.columns)):
    if(re.search("^Unnamed",df.columns[m]) != None):
        var.append(df.columns[m])
var
```

De esta forma tengo todas las columnas que quiero eliminar en una sola variable, de forma que se puede eliminar de manera fácil haciendo uso de la función *.drop*. Cabe destacar, que la primer fila también contiene datos basura, el cual son las unidades, de la misma manera se elimina utilizadno la función *.drop*.

Ahora para trabajar con los datos de manera temporal, haciendo uso de la columna *DATA* se crea una nueva llamada *Data* con los datos temporales de *DATA* pero en formato de fecha, con ayuda de pandas *.to_datetime*. Se hace esto para poder filtrar los datos de fecha y usar los datos del año 2019 nada más.

Una vez hecho esto, se pasa a convertir las listas en flotantes, ya que por defecto casi todas las columnas tienen el formato de objeto debido a que

se encuentra algun *NAN* en alguna fila de la columna. Esto se realiza con cuidado, ya que como hay unas columnas que se tienen que conservar su formato de *datetime64*, por lo que se guardan en variables para que después de la transición a flotante no se pierdan los datos temporales.

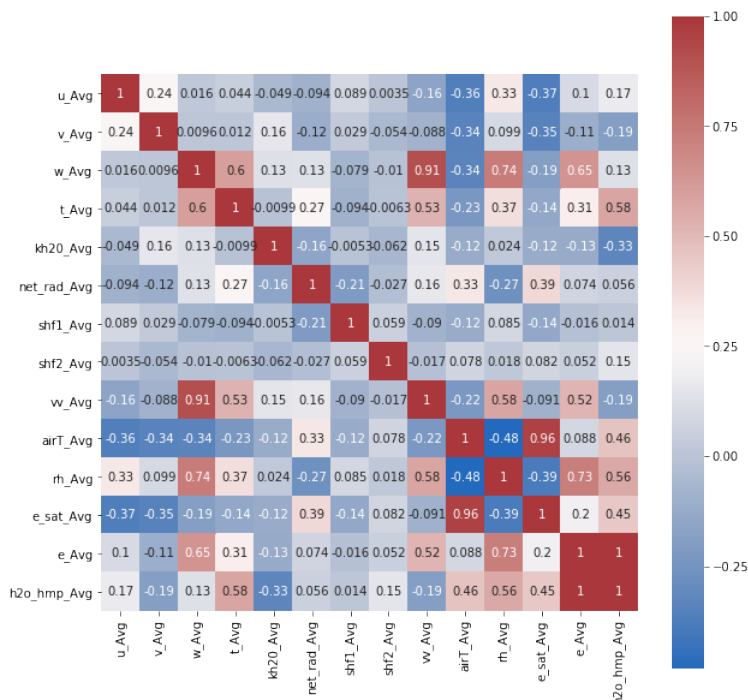
```
tiempo = df1["Date"]
tiempo2 = df1["TIME"]
df1[df1.columns] = df1[df1.columns].apply(pd.to_numeric, errors="coerce")
df1["Date"] = tiempo
df1["TIME"] = tiempo2
df1.dtypes
```

Ahora solo basta con hacer uso de la función *.corr* para hacer una tabla de comparación sobre las columnas.

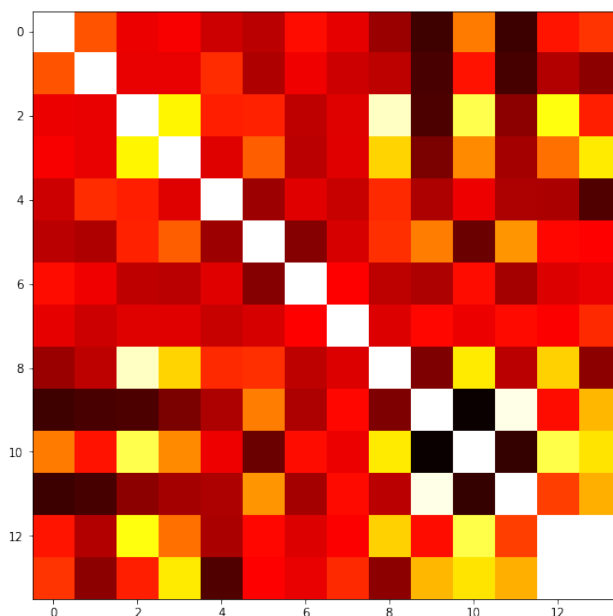
dara = df1.corr()

2. Heat-Map

Gracias a la biblioteca *seaborn* podemos hacer uso de la función *heatmap* para crear un heatmap en base a una tabla de correlación como la que nos genera la función *.corr*. Lo que nos da como resultado



Mientras que con *matplotlib* la cosa se complica más que con *seaborn* ya que se tiene que ser más específico de lo que se es con *seaborn*. He aquí un resultado de un heatmap por parte de matplotlib siendo bastante sencillo en la sección de código.



En ambos casos se trató de ser el código más sencillo posible, y claramente se ve cómo en seaborn se aprecia una mejor interfaz que con matplotlib.

3. Comparación con scatterplot

Aquí se necesitó del uso de una for que obtuviera los pares de columnas a compararse y otro for que ploteara los scatterplot de todos los pares de columnas obtenidos en el for pasado, de forma que los resultados son los siguientes:

