Web scraping: PRACTICA 1

Autor: Josep Carles Colomina Perat

Noviembre 2019

Título del dataset.

Captación horaria de datos meteorológicos entorno a una ubicación geográfica concreta.

Objetivos y contexto de la práctica.

En nuestra empresa actualmente, estamos instalando en la cubierta de la nave de nuestras instalaciones, placas solares con el objeto de realizar autoconsumo de energía eléctrica. Este autoconsumo nos permitirá por un lado reducir los costes de la energía eléctrica consumida y por otro lado, reducir el impacto de la huella de carbono de nuestras actividades.

La instalación viene equipada con equipos de control que nos permitirán conocer en tiempo real los Kwh generados, así como otros datos de interés, Kwp (pico de potencia diario), alertas de avería en el caso de que una placa sufra una avería, etc.

A pesar de ello, con el tiempo la instalación sufre progresivamente una pérdida de potencia creciente, a causa de la acumulación de suciedad, polvo, excrementos de pájaros, plásticos u otros objetos arrastrados por el viento que se depositan sobre las placas, etc. El único modo de recuperar la potencia perdida, es supervisar por periódicamente de forma visual toda la instalación y proceder a su limpieza manual si se detectan anomalías, y realizar periódicamente mantenimientos preventivos consistentes en limpiezas completas de toda la instalación, las cuales se realizan habitualmente al menos una vez al año. Una excesiva acumulación de suciedad puede llegar a requerir de limpiezas adicionales, si la potencia perdida es relevante.

Controlar la variación de potencia pico diaria, o las variaciones de Kwh producidos en un intervalo de tiempo, no resulta de mucha utilidad para determinar si es necesaria una supervisión o limpieza adicionales, ya que las variaciones climáticas que se producen en cortos períodos de tiempo (horas, días, etc,) o estacionalmente pueden ser mucha magnitud. por ejemplo el nivel de nubosidad horario puede afectar significativamente al total de Kwh producidos a lo largo de un día.

Un modo alternativo de supervisar esta pérdida de eficiencia por suciedad de la instalación, podría ser, partiendo de los datos obtenidos de estaciones de la red meteorológica cercanas a la instalación, que incluyan intensidad de radiación solar, buscar algún tipo de correlación entre estos datos meteorológicos obtenidos de estas estaciones y los datos horarios de energía producidos por la instalación, de este modo una vez obtenida esta correlación, se podrían detectar para todo tipo de

condición climática (de luz), el % de perdida de potencia admisible que está dentro de normalidad, para estas condiciones, si durante varios días se detecta un exceso de pérdida de potencia, en base a los datos de las estaciones, se procedería a supervisar la instalación, e incluso a limpiar parte de la misma si el % de pérdida es suficiente. Como he comentado esta detección de exceso de pérdida de potencia se podría detectar en cualquier estación del año, y con cualquier climatología.

La captación de datos se ha realizado directamente desde la página http://www.meteoclimatic.net, para ello primero se han localizado al menos las dos estaciones más cercanas a la instalación donde se inatalarán las placas solares, de modo que al menos una de ellas disponga de un equipo de medición de radiación solar, ya que esta variable es fundamental para poder determinar la potencia que generará la instalación. Estos equipos de medición no son muy habituales, en nuestro caso una de las dos estaciones más cercana disponía de este equipo.

En esta página web, cada estación está claramente identificada por un código y es fácil acceder por url, a los datos de las mismas, en contreto las estaciones desde las que se captarán los datos son las de Olesa de Montserrat y la de Vacarisses, ubicadas las dos cerca de Barcelona.

El código de captación es bastante modular, por lo que añadir más estaciones a la captación de datos no se ha considerado necesario a efectos de la práctica.

Destacar que la mayoría de las captaciones de datos presentados en el archivo csv, se han realizado a lastir de las 18:00 horas, por lo que los valores de radiación solar a partir de las mismas siempre tendrán un valor de 0 W/m2.

Descripción del dataset y contenido del mismo, descripción de los datos del dataset.

El dataset recoge registros tomados a una fecha y hora determinada (hora servidor donde se ejecuta el script), y cada registro horario recoge los datos meteoclimáticos de las dos estaciones indicadas, en concreto estos parámetros son, temperatura, % de humedad, velocidad del viento, dirección del viento (enla estación), presión, radiación solar y precipitación.

El script es capaz una vez creado el archivo csv, de captar varios o muchos valores horarios, indicando el número de ciclos que se quieren registrar. La captación se realiza en secuencias temporales de 1 hora. Una vez ejecutados los ciclos indicados, el script se puede reiniciar a voluntad para seguir anotando registros en el mismo archivo csv creado inicialmente.

Los datos se han ido tomando diariamente en ciclos de 4-5 horas diarias, casi siempre por las tardes a partir de las 18 h., por motivos laborales, ya que por las mañanas no me ha resultado posible ejecutar el cript. Este punto es relevante ya que la radiación solar es un dato importante para este dataset, y por las tardes en esta época del año, su valor es siempre 0, los valores distintos de cero han sido tomados los fines de semana por la mañana. Para esta práctica, estos valores 0, no se

consideran relevante a efectos didácticos, ya que el script puede ejecutarse en un servidor funcionando en continuo captando datos las 24 h.

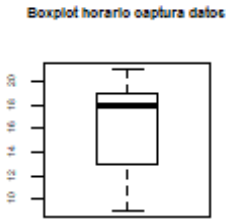
Dado que los datos se toman cada hora, no hay riesgo de que el script tenga incidencia o ralentice los servidores de la web en la que se captan los datos.

La descripción de las variables es la siguiente:

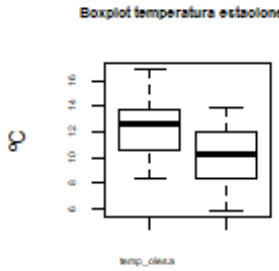
- Fecha, feca en la que se recopilan los registros.
- Hora, dado que los Kwh producidos por la instalación se recuperarán con granularidad horaria (solamente indicando la hora de producción, no se especificarán minutos), la hora recopilada no será la de medición en la estación, si no la hora de reloj del servidor que ejecuta el script. Dado que las variaciones de las variables climáticas varian lentamente con el tiempo, por lo que recuperar la hora y minutos de la captación del dato en la estación, considero no es relevante.
- Temperatura, dato de interés sobre las condiciones generales climáticas, la T ambiente también puede tener un ligero efecto sobre la producción de Kwh en las placas solares.
- % de humedad, puede indicar condiciones lluviosas, además bajas temperaturas junto con % de humedad elevadas incrementan la condensación de gotas de agua sobre la superficie de las placas, lo cual puede reducir su eficiencia.
- Velocidad del viento en Km/h, esta variable es posible que sea poco relevante, aunque puede indicar condiciones climáticas inestables (menos radiación solar incidente).
- Dirección del viento, también se espera que sea poco relavante, aunque pude aportar información geográfica de interés, ya que al tratarse de una comarca con una orografía compleja, determinadas direcciones de viento pueden implicar el arrastre de partículas en suspensión que pueden ensuciar las placas, por ejemplo si las masas de aire proceden de zonas industriales, o de núcleos urbanos importantes.
- Presión en mmHg, es indicativa de condiciones meteoclimáticas inestables y por lo tanto indicativas de mayor o menor presencia de masa nubosas.
- Radiación solar en W/m2, este parámetro es muy importante ya que está correlacionado linealmente con la producción de Kwh de la instalación fotovoltáica, cuanto más cerca esté de las instalaciones mejor.
- Precipitación en mm, cuanto mayor sea este valor menor será la producción en Kwh de la instalación.

Representación gráfica.

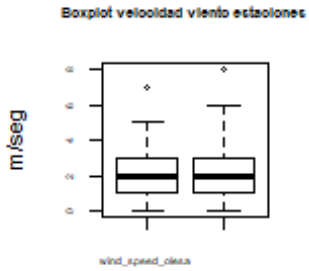
Los datos obtenidos pueden ser fácilmente representados por días de captura y por variable, lo cual nos permitirá seguir los valores de los diferentes parámetros, en las series temporales en las que se hayan captado estos, a modo de ejemplo podemos ver :



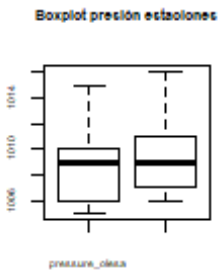
Hora



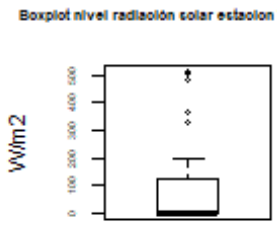
T



Velocidad

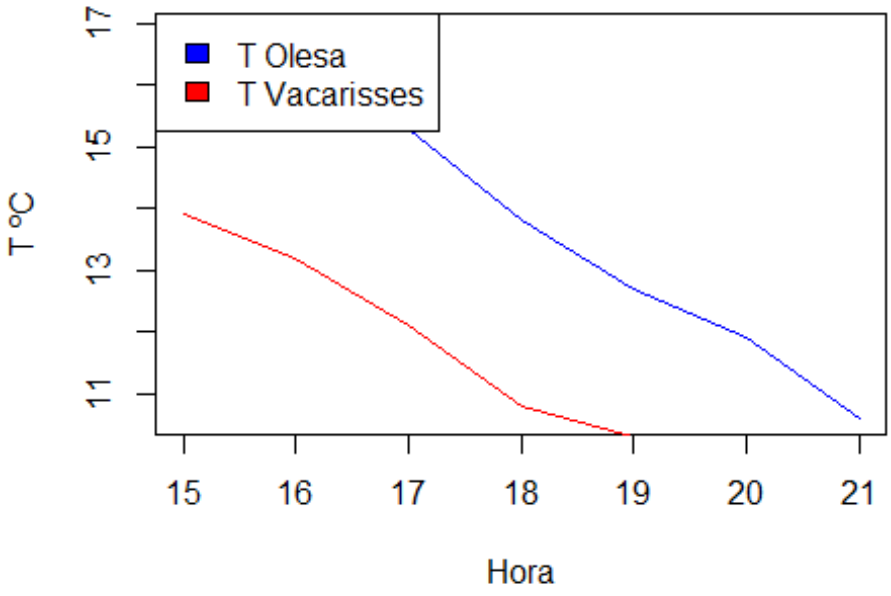


Presión



Radiacion solar

Datos de temperatura estaciones día 27-11-2019



Agradecimientos.

Los datos de esta práctica han sido tomados en http://www.meteoclimatic.net, Meteoclimatic es una gran red de estaciones meteorológicas automáticas no profesionales en tiempo real y un importante directorio de recursos meteorológicos. Los observatorios del ámbito de cobertura de Meteoclimatic: Península Ibérica, los dos archipiélagos, sur de Francia y la África cercana al Estrecho de Gibraltar.

Inspiración.

Tal y como se ha explicado en los objetivos, la inspiración de esta práctica es captar datos meteorológicos de estaciones, con el objetivo de realizar una evaluación predictiva que permita detectar pérdidas de eficiencia de una instalación fotovoltáica, en base a la comparación entre los Kwh producidos por la instalación y los datos meteorológicos medidos en su entorno.

Licencia.

Código del programa.

```
##########################
#Función captura de datos#
##########################
def captura():

    page_olesa =
requests.get("https://www.meteoclimatic.net/perfil/ESCAT0800000008640A"
)
    soup_olesa = BeautifulSoup(page_olesa.content)

    page_vacarisses =
requests.get("https://www.meteoclimatic.net/perfil/ESCAT0800000008233A"
)
    soup_vacarisses = BeautifulSoup(page_vacarisses.content)

    #Buscamos los datos de interés que en la página se encuentran en el

    #'td' de clase "dadesactuals".
    #Seleccionamos todos los datos menos el último, ya que los días de

    #sequía no nos interesan.
    dades_olesa = soup_olesa.find_all('td', class_="dadesactuals")[0:5]
    dades_vacarisses = soup_vacarisses.find_all('td',
class_="dadesactuals")[0:6]
```

```
    #Generamos los vectores para almacenar los datos.
    meteo_dat_olesa = []
    meteo_dat_vacarisses = []

    #Almacenamos los strings en los vectores de datos.
    for elem in dades_olesa:
        meteo_dat_olesa.append(elem.get_text())
    for elem in dades_vacarisses:
        meteo_dat_vacarisses.append(elem.get_text())

    #Añadimos dos posiciones más a la lista ya que la dirección del

    #viendo se descompone en dos variables.
    meteo_dat_olesa.insert(2,None)
    meteo_dat_vacarisses.insert(2,None)

    #Modificamos ahora cada uno de los strings para convertirlos en
datos numéricos.
    meteo_dat_olesa[0] = float(meteo_dat_olesa[0][:4])
    meteo_dat_vacarisses[0] = float(meteo_dat_vacarisses[0][:4])
    meteo_dat_olesa[1] = int(meteo_dat_olesa[1][:2])
    meteo_dat_vacarisses[1] = int(meteo_dat_vacarisses[1][:2])

    #En la dirección del viento, el el segundo char aparece como \xa0
(espacio vacío)#
    #por ejemplo en N (N\xa0), este espacio vacío se corresponde con el
caracter chr(160)#
    if meteo_dat_olesa[3][1:2] == chr(160):
        meteo_dat_olesa[2] = int(meteo_dat_olesa[3][3:4])
    elif meteo_dat_olesa[3][2:3] == chr(160):
        meteo_dat_olesa[2] = int(meteo_dat_olesa[3][4:5])
    else:
        meteo_dat_olesa[2] = int(meteo_dat_olesa[3][5:6])

    if meteo_dat_vacarisses[3][1:2] == chr(160):
        meteo_dat_vacarisses[2] = int(meteo_dat_vacarisses[3][3:4])
    elif meteo_dat_vacarisses[3][2:3] == chr(160):
        meteo_dat_vacarisses[2] = int(meteo_dat_vacarisses[3][4:5])
    else:
        meteo_dat_vacarisses[2] = int(meteo_dat_vacarisses[3][5:6])

    #En la dirección del viento, el el segundo char aparece como \xa0
(espacio vacío)#
    #por ejemplo en N (N\xa0), este espacio vacío se corresponde con el
caracter chr(160)#
    if meteo_dat_olesa[3][1:2] == chr(160):
        meteo_dat_olesa[3] = meteo_dat_olesa[3][0:1]
    elif meteo_dat_olesa[3][2:3] == chr(160):
```

```
        meteo_dat_olesa[3] = meteo_dat_olesa[3][0:2]
    else:
        meteo_dat_olesa[3] = meteo_dat_olesa[3][:3]
    if meteo_dat_vacarisses[3][1:2] == chr(160):
        meteo_dat_vacarisses[3] = meteo_dat_vacarisses[3][0:1]
    elif meteo_dat_olesa[3][2:3] == chr(160):
        meteo_dat_vacarisses[3] = meteo_dat_vacarisses[3][0:2]
    else:
        meteo_dat_vacarisses[3] = meteo_dat_vacarisses[3][:3]
    meteo_dat_olesa[4] = int(meteo_dat_olesa[4][:4])
    meteo_dat_vacarisses[4] = int(meteo_dat_vacarisses[4][:4])
    meteo_dat_olesa[5] = float(meteo_dat_olesa[5][:3])
    if meteo_dat_vacarisses[5][2:3] == 'W':
        meteo_dat_vacarisses[5] = int(meteo_dat_vacarisses[5][:1])
    else:
        meteo_dat_vacarisses[5] = int(meteo_dat_vacarisses[5][:3])
    meteo_dat_vacarisses[6] = float(meteo_dat_vacarisses[6][:3])
    return meteo_dat_olesa + meteo_dat_vacarisses

a = captura()
print(a)



#########
# Main #
########

import requests
from bs4 import BeautifulSoup
import time
import datetime
from datetime import date
from datetime import datetime
import datetime
import csv

#Listado de atributos del csv.
names = ['dia', 'hora', 'temp_olesa', '%_hum_olesa',
'wind_speed_olesa','dir_wind_speed_olesa', 'pressure_olesa',
'rain_olesa',
'temp_vacarisses', '%_hum_vacarisses',
'wind_speed_vacarisses', 'dir_wind_speed_vacarisses',
'pressure_vacarisses',
'radiation_vacarisses', 'rain_vacarisses']

#Generamos el csv por primera vez.
sn = str(input("¿Desea inicializar el archivo CSV de captura de datos
s/n? "))
```

```
if sn == 's':
    csvsalida = open('meteo_data.csv', 'w', newline='')
    salida = csv.writer(csvsalida)
    salida.writerow(names)
    del salida
    csvsalida.close()
else:
    pass

#Número de ciclos horarios en los que se registrarán los datos.
ciclos = int(input("¿Durante cuántas horas quiere que se registren los
datos? "))
#Abrimos de nuevo el archivo csv para continuar archivando datos.
csvsalida = open('meteo_data.csv', 'a', newline='')
salida = csv.writer(csvsalida)
#En cada ciclo realizaremos una lectura en las estaciones.
for i in range(ciclos):
    #Unimos los listados de fecha y hora junto con los datos obtenidos

    #de las estaciones mediante la función capture().
    data_register = [datetime.date.today(),
    datetime.datetime.now().hour] + captura()
    print(data_register)
    salida.writerow(data_register)
    #Con time.sleep aplazamos cada lectura 1 hora (3600 segundos).
    time.sleep(3600)

del salida
csvsalida.close()
```

Dataset capturado.

```
data <- read.csv("meteo_data.csv", header = TRUE, sep = ",")
data
```

```
##           dia hora temp_olesa X._hum_olesa wind_speed_olesa
## 1  2019-11-06   17       16.3           55                2
## 2  2019-11-06   18       14.9           59                1
## 3  2019-11-06   19       13.3           64                0
## 4  2019-11-06   20       12.2           68                0
## 5  2019-11-07   15       16.9           57                2
## 6  2019-11-07   16       16.3           57                3
## 7  2019-11-07   17       15.3           58                2
## 8  2019-11-07   18       13.8           60                0
## 9  2019-11-07   19       12.7           59                3
## 10 2019-11-07   20       11.9           59                1
## 11 2019-11-07   21       10.6           63                0
## 12 2019-11-08   17       12.9           52                7
## 13 2019-11-08   18       11.3           59                5
```

```
## 14 2019-11-08   19       10.3           63                1
## 15 2019-11-08   20        9.8           64                7
## 16 2019-11-09    9        8.8           70                3
## 17 2019-11-09   10       10.5           65                5
## 18 2019-11-09   11       12.3           60                4
## 19 2019-11-09   18       12.9           60                1
## 20 2019-11-09   19       13.3           57                3
## 21 2019-11-09   20       13.1           57                1
## 22 2019-11-10    9        8.3           88                0
## 23 2019-11-10   10        9.3           77                0
## 24 2019-11-10   11       11.6           68                2
## 25 2019-11-10   13       15.0           44                5
##    dir_wind_speed_olesa pressure_olesa rain_olesa temp_vacarisses
## 1                   OSO           1009        0.0            13.6
## 2                   OSO           1010        0.0            12.7
## 3                     S           1010        0.0            12.0
## 4                     S           1010        0.0            11.6
## 5                   NNO           1005        0.0            13.9
## 6                   NNE           1006        0.0            13.2
## 7                   NNO           1005        0.0            12.1
## 8                   ONO           1006        0.0            10.8
## 9                   OSO           1006        0.0            10.3
## 10                   SE           1006        0.0             9.9
## 11                    N           1006        0.0             9.3
## 12                    O           1007        0.0            10.0
## 13                    O           1007        0.0             8.4
## 14                  NNO           1007        0.0             7.5
## 15                  NNO           1008        0.0             7.2
## 16                    N           1015        0.0             5.8
## 17                    N           1015        0.0             7.7
## 18                    N           1015        0.0             9.5
## 19                    O           1012        0.0            10.4
## 20                    O           1011        0.0            10.3
## 21                  SSO           1010        0.0            10.2
## 22                  SSO           1009        3.4             6.4
## 23                  OSO           1009        3.4             7.0
## 24                    N           1009        3.4             9.3
## 25                    N           1007        3.4            12.2
##    X._hum_vacarisses wind_speed_vacarisses dir_wind_speed_vacarisses
## 1                 62                     3                       ONO
## 2                 64                     1                       NO
## 3                 65                     6                       NNE
## 4                 67                     8                         N
## 5                 65                     6                       OSO
## 6                 67                     3                         O
## 7                 68                     1                       NO
## 8                 67                     1                         O
## 9                 66                     6                       NO
## 10                61                     1                         N
## 11                61                     2                       ONO
```

```
## 12                61                       2                       ONO
## 13                67                       2                         O
## 14                71                       8                       NNO
## 15                71                       1                       ONO
## 16                80                       1                       NNE
## 17                74                       2                       NO 
## 18                69                       1                       NNO
## 19                67                       1                       NNO
## 20                67                       3                       NNO
## 21                66                       1                       ONO
## 22                90                       0                       NNE
## 23                85                       3                       NNE
## 24                75                       1                         N
## 25                51                       8                         N
##    pressure_vacarisses radiation_vacarisses rain_vacarisses
## 1                 1010                     5              0.0
## 2                 1011                     0              0.0
## 3                 1011                     0              0.0
## 4                 1011                     0              0.0
## 5                 1007                   332              0.0
## 6                 1007                    86              0.0
## 7                 1006                    40              0.0
## 8                 1006                     0              0.0
## 9                 1006                     0              0.0
## 10                1007                     0              0.0
## 11                1007                     0              0.0
## 12                1008                   111              0.0
## 13                1008                     0              0.0
## 14                1008                     0              0.0
## 15                1009                     0              0.0
## 16                1016                    63              0.0
## 17                1016                   367              0.0
## 18                1016                   480              0.0
## 19                1013                     0              0.0
## 20                1012                     0              0.0
## 21                1011                     0              0.0
## 22                1010                   127              2.4
## 23                1010                   195              2.4
## 24                1009                   506              2.4
## 25                1008                   510              2.4
```