UNIVERSIDAD COMPLUTENSE DE MADRID

ntic master
revolucionamos la comunicación

Master Big Data y Data Science

Aplicaciones al comercio, empresa y finanzas

Programación Python

(avanzado)

PRÁCTICA

[redacted]

Introducción

La práctica del módulo “Programación Python (avanzado)” servirá para afianzar y confirmar que se ha adquirido el conocimiento de la materia impartida a lo largo del módulo. Gran parte del conocimiento necesario para realizar esta práctica se ha explicado y desarrollado en clase mediante ejemplos que se pueden reutilizar. Ante cualquier duda sobre cómo realizar algunos de los puntos se puede utilizar la documentación entregada al inicio del módulo en la que vienen diferentes ejemplos y que el alumno podrá apoyarse en ellos.

¿En qué consiste la práctica?

El alumno, empleando los conocimientos adquiridos en los módulos de “Fundamentos de Python” y “Python Avanzado”, debe ser capaz de generar un Dataset (conjunto de datos) a partir de un algoritmo que genere nombres de servidores (*hostnames*) “random” y que habremos visto previamente en clase. Con este Dataset el alumno debe ser capaz de generar un DataFrame de Pandas y debe generar una serie de gráficos con Matplotlib. No es obligatorio realizar todos los puntos. Se puede hacer una entrega parcial, realizando solo los puntos que el alumno pueda o sepa hacer. Se considerará aprobada si la nota es igual o mayor a 5 puntos.

Requisitos y puntuaciones para realizar la práctica:

1. Importar todas las librerías necesarias (0.15 puntos)

2. Inicializar algunas variables que después modificaremos (0.15 puntos)

3. Crear una función para generar los hostnames en base a unas reglas (1.5 puntos)

   Estas reglas son:

   - La función se ha de llamar `set_hostnames` y debe recibir un parámetro llamado `number_of_hosts` de tipo `int` que represente el número de hosts que queremos generar.
   - El hostname debe estar compuesto por un total de 8 carácteres alfanuméricos, las letras siempre mayúsculas.
   - El primer caracter debe indicar el sistema operativo, siendo `L` para `Linux`, `S` para `Solaris`, `A` para `AIX` y `H` para `HP-UX`. La proporción aproximada de sistemas operativos debe ser:

    - **Linux**: 40%
    - **Solaris**: 30%
    - **AIX**: 20%
    - **HP-UX**: 10%
- El segundo caracter debe indicar el entorno, siendo D para Development, I para Integration, T para Testing, S para Staging y P para Production. La proporción aproximada de entornos debe ser:
    - **Development**: 10%
    - **Integration**: 10%
    - **Testing**: 25%
    - **Staging**: 25%
    - **Production**: 30%
- Los tres siguientes caracteres deben indicar el país, siendo NOR para Norway, FRA para France, ITA para Italy, ESP para Spain, DEU para Germany e IRL para Ireland. La proporción aproximada de países debe ser:
    - **Norway**: 6%
    - **France**: 9%
    - **Italy**: 16%
    - **Spain**: 16%
    - **Germany**: 23%
    - **Ireland**: 30%
- Por último 3 dígitos que indiquen el número de nodo que ya existe para un mismo sistema operativo, entorno y país. El valor debe ser incremental, comenzando en 001 y con un valor máximo de 999.

4. Crear una función para obtener el nombre del SO (0.5 puntos)

   La función se ha de llamar get_os, debe recibir un parámetro llamado hostname de tipo str y debe devolver una cadena Linux, Solaris, AIX o HP-UX dependiendo de la primera letra del parámetro hostname. Debería ser improbable que el hostname recibido como parámetro comience por una letra diferente de L, S, A o H, pero de darse el caso, la función debe devolver la cadena Unknow.

5. Crear una función para obtener el nombre del entorno (0.5 puntos)

   La función se ha de llamar get_enviroment, debe recibir un parámetro llamado hostname de tipo str y debe devolver una cadena Development, Integration, Testing, Staging o Production dependiendo

de la segunda letra del parámetro hostname. Debería ser improbable que el hostname recibido como parámetro tenga por segundo caracter por una letra diferente de D, I, T, S o P, pero de darse el caso, la función debe devolver la cadena Unknow.

6. Creamos una función para obtener el nombre del país (0.5 puntos)

   La función se ha de llamar get_country, debe recibir un parámetro llamado hostname de tipo str y debe devolver una cadena Norway, Germany, Italy, Spain, Ireland o France dependiendo de las letras de la tercera a la quinta del parámetro hostname. Debería ser improbable que el hostname recibido como parámetro tenga por caracteres en las posiciones 3, 4, 5, caracteres diferentes de NOR, DEU, ITA, ESP, IRL o FRA, pero de darse el caso, la función debe devolver la cadena Unknow.

7. Crear una función para generar el DataFrame (1 punto)

   La función se ha de llamar set_dataframe y debe recibir un parámetro llamado count de tipo int, que represente el número de registros (filas) que vemos a generar. Para poder establecer un valor a la variable df que se encuentra fuera de la función y que inicialmente iniciamos con un valor None, debemos invocar a la variable df como global dentro de esta función. A continuación debemos llamar a la función set_hostnames pasándole como argumento el parámetro count. Después debemos ir añadiendo a la lista dataset que teníamos inicializada al principio como lista vacía [] un diccionario por cada hostname de nuestra lista hostnames. Los campos de este diccionario deben ser:
   - **hostname**: Por ejemplo LDIRL003.
   - **os**: Por ejemplo Linux.
   - **enviroment**: Por ejemplo Development.
   - **country**: Por ejemplo Ireland.
   - **node**: De tipo int, por ejemplo 3.

   Finalmente creamos un DataFrame utilizando los datos de la lista de diccionarios dataset y asignando el DataFrame de Pandas a la variable global df.

8. Crear el DataFrame (0.2 puntos)

   Invocamos a la función set_dataframe pasándole como argumento el

entero 1500. Inspeccionamos el DataFrame df para ver si se ha generado bien.

9. Guardar el DataFrame generado en un fichero CSV (0.5 puntos)

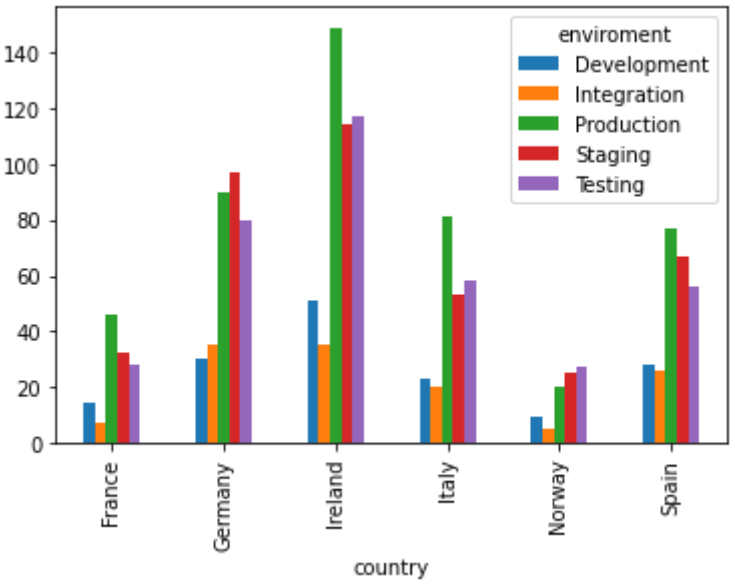
   El dataframe df recien generado debemos volcarlo a un fichero CSV llamado hosts.csv, ubicado en la misma carpeta donde se encuentra el libro de Jupyter Notebook, debe incluir las cabeceras (header=True) y no debe incluir los índices (index=False). A continuación hay que hacer la prueba de leer el archivo generado mediante el método read_csv, almacenar el DataFrame en una variable llamada hosts_df y visualizarlo para ver si se ha generado bien. Se tiene que ver así (*más o menos, los nombres de los hostnames evidentemente no tienen por qué ser exactamente iguales*):

	hostname	os	enviroment	country	node
0	LTIRL001	Linux	Testing	Ireland	1
1	HSIRL001	HP-UX	Staging	Ireland	1
2	ATIRL001	AIX	Testing	Ireland	1
3	ASDEU001	AIX	Staging	Germany	1
4	STIRL001	Solaris	Testing	Ireland	1
...	...	...	...	...	...
1495	LSDEU044	Linux	Staging	Germany	44
1496	SSIRL027	Solaris	Staging	Ireland	27
1497	SPFRA015	Solaris	Production	France	15
1498	APFRA013	AIX	Production	France	13
1499	LSESP018	Linux	Staging	Spain	18

1500 rows × 5 columns

10. Generar un único gráfico agrupando para cada país (country) los entornos (enviroment) (0.5 puntos)

    Se debe utilizar la función unstack y se debe generar un plot de tipo barras (kind=bar). Debe quedar (más o menos) así:



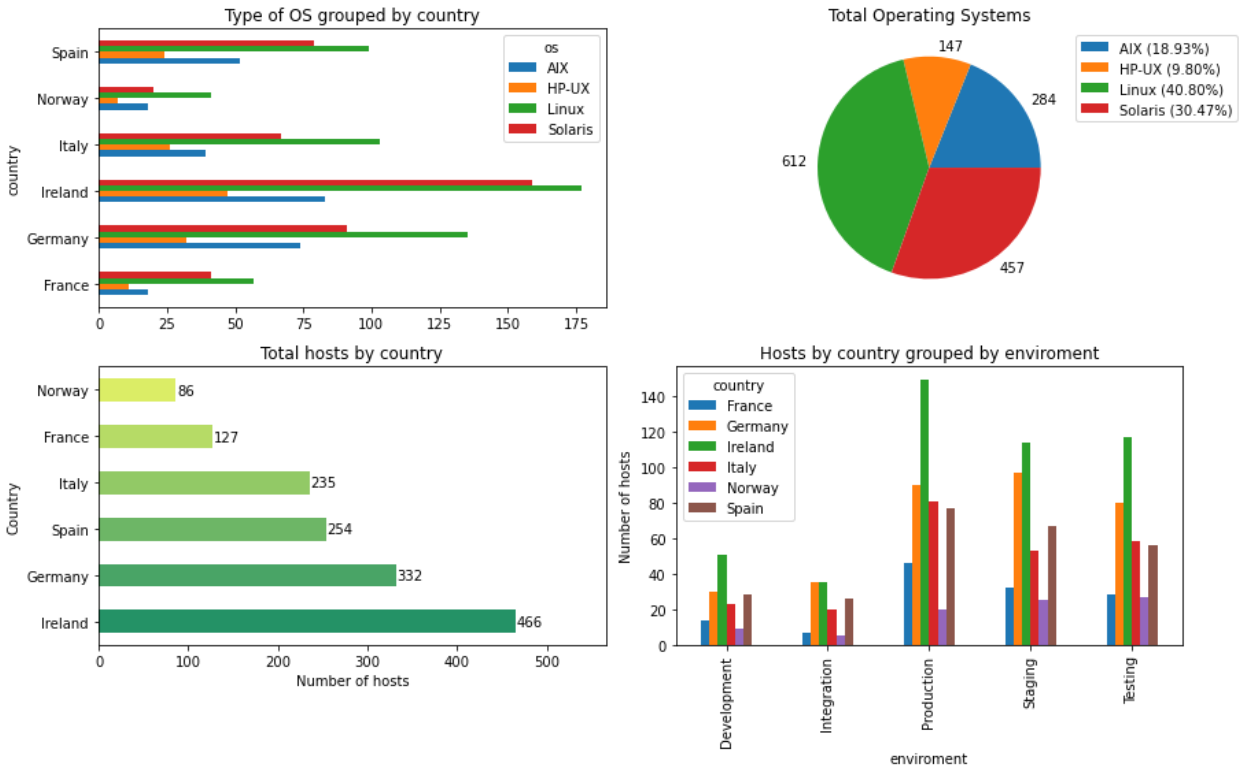
11. Crear una figura con 4 gráficos en una malla de 2 filas y 2 columnas (4.5 puntos)

    - En la esquina superior izquierda debe aparecer un gráfico cuyo título sea Type of OS grouped by country. Debe ser un gráfico de barras horizontales que representen una agrupación (groupby) por cada país (country) de los sistemas operativos (os) que tiene. Se debe utilizar la función unstack y el plot debe ser de tipo barras horizontales (barh).
    - En la esquina superior derecha debe aparecer un gráfico cuyo título sea Total Operating Systems. Debe representar la cantidad total de sistemas operativos (os) que hay en el DataFrame. Se debe utilizar la función groupby y el gráfico debe ser de tipo tarta (pie). Como etiquetas (labels) debe mostrar el número de sistemas operativos de cada tipo, y además debe mostrarse una leyenda (legend) en la esquina superior derecha en la que aparezca para cada sistema operativo el porcentaje existente en el DataFrame.
    - En la esquina inferior izquierda debe aparecer un gráfico cuyo título sea Total hosts by country. Debe ser un gráfico de barras horizontales que representen la cantidad total de hosts por cada país, para ello se debe utilizar la función value_counts() sobre los países (country) del DataFrame. El gráfico generado debe incluir como etiqueta en el eje x el texto Number of hosts y como etiqueta del eje y el texto Country. También se ha de incluir el

número total de hosts que tiene cada país a la derecha de cada barra horizontal. Además, se ha de añadir como valor máximo del eje x un número equivalente al número total de hosts+100, de este modo se verá un pequeño margen a la derecha que hará que se visualice un poco mejor. Opcionalmente (si se hace puntuará) se puede añadir con la librería seaborn una paleta de colores (color_palette) que podemos utilizar para darle un color degradado a las barras.

- En la esquina inferior derecha debe aparecer un gráfico cuyo título sea Hosts by country grouped by enviroment. Debe representar una agrupación (groupby) de hosts que hay por cada país (country) y entorno (enviroment). Se debe utilizar la función unstack(0) y el plot debe ser de tipo barras (bar). Como etiqueta del eje y se debe añadir el texto Number of hosts.
  Finalmente se deben ajustar los márgenes y espacios entre los gráficos (fig.tight_layout()).

Debe quedar (más o menos) así:



Entrega de la práctica una vez finalizada

La práctica debe realizarse obligatoriamente en **Jupiter Notebook** y debe guardarse en un archivo que tenga el siguiente nombre y extensión:

DNI_NOMBRE_APELLIDO_PRACTICA_PYTHON_AVANZADO.ipynb

Por ejemplo, si el nombre del alumno es *John Doe* y su DNI es *12345678A* el archivo que hay que generar y entregar debe tener el siguiente nombre:

12345678A_JOHN_DOE_PRACTICA_PYTHON_AVANZADO.ipynb

No se debe adjuntar ninguna otra documentación adicional. Cualquier duda sobre la realización de la práctica se puede consultar en el foro del módulo o contactar con el profesor a través de la plataforma de la Universidad.